Métodos básicos

En la parte 1, exploramos el entorno de R y discutimos cómo ingresar datos de una amplia variedad de fuentes, combinarlos y transformarlos, y prepararlos para análisis adicionales. Una vez que sus datos se han ingresado y limpiado, el siguiente paso suele ser explorar las variables una a la vez. Esto le proporciona información sobre la distribución de cada variable, que es útil para comprender las características de la muestra, identificar valores inesperados o problemáticos y seleccionar los métodos estadísticos apropiados. A continuación, las variables se estudian típicamente de dos en dos. Esto puede ayudarlo a descubrir relaciones básicas entre las variables y es un primer paso útil para desarrollar modelos más complejos.

La Parte 2 se centra en técnicas gráficas y estadísticas para obtener información básica sobre los datos. El capítulo 6 describe los métodos para visualizar la distribución de variables individuales. Para las variables categóricas, esto incluye gráficos de barras, gráficos circulares y el gráfico de ventilador más nuevo. Para las variables numéricas, esto incluye histogramas, gráficos de densidad, diagramas de caja, diagramas de puntos y el gráfico de violín menos conocido. Cada tipo de gráfico es útil para comprender la distribución de una sola variable.

El capítulo 7 describe métodos estadísticos para resumir variables individuales y relaciones bivariadas. El capítulo comienza con la cobertura de estadísticas descriptivas para datos numéricos basados en el conjunto del conjunto de datos y en subgrupos de interés. A continuación, se describe el uso de tablas de frecuencias y tabulaciones cruzadas para resumir datos categóricos. El capítulo termina discutiendo métodos inferenciales básicos para comprender las relaciones entre dos variables a la vez, incluidas las correlaciones bivariadas, las pruebas de chi cuadrado, las pruebas t y los métodos no paramétricos.

Cuando haya terminado esta parte del libro, podrá utilizar métodos gráficos y estadísticos básicos disponibles en R para describir sus datos, explorar las diferencias de grupo e identificar relaciones significativas entre las variables.

Capítulo 6. Gráficos básicos

Este capítulo cubre

- Gráficos de barras, cajas y puntos
- Gráficos circulares y de fans
- Histogramas y diagramas de densidad de núcleo

Siempre que analizamos datos, lo primero que debemos hacer es mirarlos. Para cada variable, ¿cuáles son los valores más comunes? ¿Cuánta variabilidad está presente? ¿Hay alguna observación inusual? R proporciona una gran cantidad de funciones para visualizar datos. En este capítulo, veremos gráficos que le ayudan a comprender una sola variable categórica o continua. Este tema incluye

- Visualización de la distribución de una variable
- Comparación de grupos en una variable de resultado

En ambos casos, la variable puede ser continua (por ejemplo, kilometraje del automóvil como millas por galón) o categórica (por ejemplo, el resultado del tratamiento como ninguno, algunos o marcado). En capítulos posteriores, exploraremos gráficos que muestran relaciones bivariadas y multivariantes entre variables.

Las siguientes secciones exploran el uso de gráficos de barras, gráficos circulares, gráficos de ventiladores, histogramas, gráficos de densidad de núcleo, gráficos de caja, gráficos de violín y diagramas de puntos. Algunos de estos pueden ser familiares para usted, mientras que otros (como tramas de ventiladores o tramas de violín) pueden ser nuevos para usted. El objetivo, como siempre, es comprender mejor sus datos y comunicar esta comprensión a los demás. Comencemos con las parcelas de bar.

6.1. Parcelas de bar

Un gráfico de barras muestra la distribución (frecuencia) de una variable categórica a través de barras verticales u horizontales. En su forma más simple, el formato de la función de gráfico de barras es ***barplot()***

```
barplot(height)
```

donde *height* es un vector o matriz.

En los siguientes ejemplos, trazará el resultado de un estudio que investiga un nuevo tratamiento para la artritis reumatoide. Los datos están contenidos en el marco de datos de Arthritis distribuido con el paquete vcd. Este paquete no se incluye en la instalación predeterminada de R, así que instálelo antes del primer uso (***install.packages("vcd")***).

Tenga en cuenta que el paquete vcd no es necesario para crear gráficos de barras. Lo está cargando para obtener acceso al conjunto de datos de Arthritis. Pero necesitará el paquete vcd al crear espinogramas, que se describen en la sección 6.1.5.

6.1.1. Parcelas de barras simples

Si la altura es un vector, los valores determinan las alturas de las barras en la gráfica y se produce una gráfica de barras vertical. La inclusión de la opción ***horiz=TRUE*** produce un gráfico de barras horizontal en su lugar. También puede agregar opciones de anotación. La opción *main* agrega un título de trama, mientras que las opciones ***xlab*** e ***ylab*** agregan etiquetas de eje x y eje y, respectivamente.

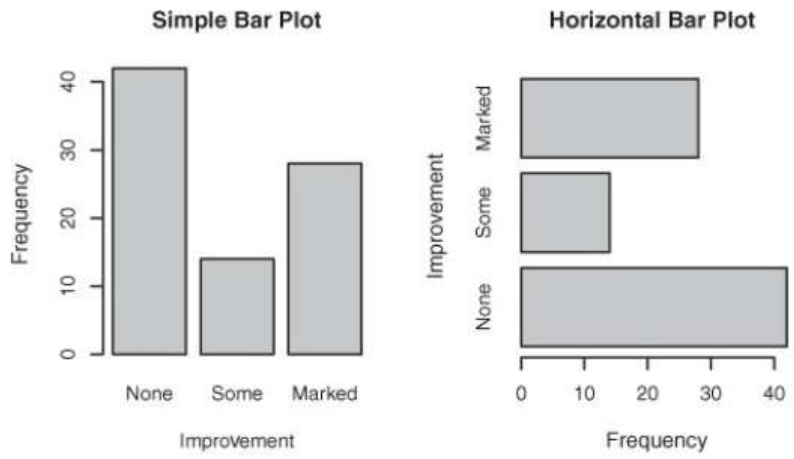
En el estudio de Arthritis, la variable Improved registra los resultados de los pacientes para las personas que reciben un placebo o medicamento:

```
library(vcd)
> counts <- table(Arthritis$Improved)
> counts
  None   Some Marked
    42     14     28
```

Aquí, se ve que 28 pacientes mostraron una mejoría marcada, 14 mostraron alguna mejoría y 42 no mostraron mejoría. Discutiremos el uso de la función ***table()*** para obtener recuentos de celdas más completamente en el capítulo 7.

Puede graficar los recuentos de variables utilizando un gráfico de barras vertical u horizontal. El código se proporciona en la siguiente lista y los gráficos resultantes se muestran en la figura 6.1.

Figura 6.1. Gráficos de barras verticales y horizontales simples



Listado 6.1. Gráficos de barras simples

```
barplot(counts,
        main="Simple Bar Plot",
        xlab="Improvement", ylab="Frequency")
barplot(counts,
        main="Horizontal Bar Plot",
        xlab="Frequency", ylab="Improvement",
        horiz=TRUE)
```

Simple bar plot

Horizontal bar plot

Creación de gráficos de barras con variables de factor

Si la variable categórica que se va a trazar es un factor o un factor ordenado, puede crear una gráfica de barras verticales rápidamente con la función ***plot()***. Debido a que ***Arthritis$Improved*** es un factor, el código

```
plot(Arthritis$Improved, main="Simple Bar Plot",
     xlab="Improved", ylab="Frequency")
plot(Arthritis$Improved, horiz=TRUE, main="Horizontal Bar Plot",
     xlab="Frequency", ylab="Improved")
```

generará los mismos gráficos de barras que los del listado 6.1, pero sin necesidad de tabular valores con la función ***table().***

¿Qué pasa si tienes etiquetas largas? En la sección 6.1.4, verá cómo ajustar las etiquetas para que no se superpongan.

6.1.2. Parcelas de barras apiladas y agrupadas

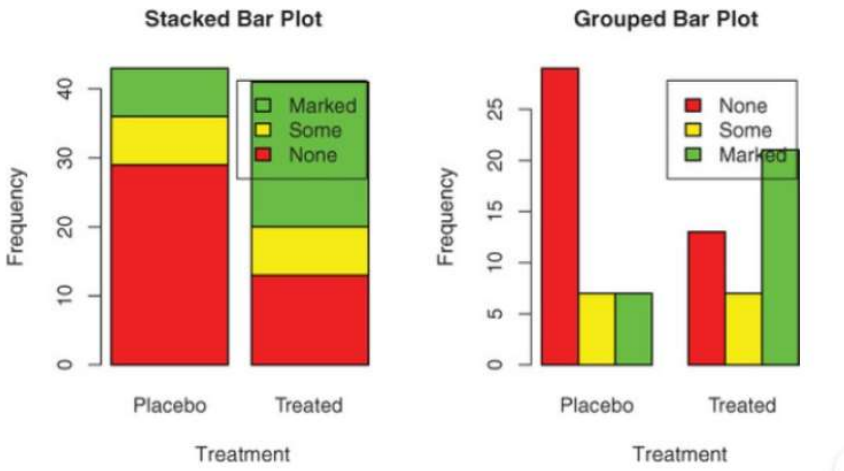
Si *height* es una matriz en lugar de un vector, el gráfico resultante será un gráfico de barras apiladas o agrupadas. Si ***beside=FALSE*** (el valor predeterminado), entonces cada columna de la matriz produce una barra en la gráfica, con los valores en la columna dando las alturas de las "subbarras" apiladas. Si *beside=TRUE*, cada columna de la matriz representa un grupo, y los valores de cada columna se yuxtaponen en lugar de apilarse.

Considere la tabulación cruzada del tipo de tratamiento y el estado de mejora:

```
> library(vcd)
> counts <- table(Arthritis$Improved, Arthritis$Treatment)
> counts
        Treatment
Improved Placebo Treated
  None        29      13
  Some         7       7
  Marked       7      21
```

Puede graficar los resultados como un gráfico de barras apiladas o agrupadas (consulte el siguiente listado). Los gráficos resultantes se muestran en la figura

6.2. Figura 6.2. Parcelas de bar apiladas y agrupadas



Listado 6.2. Parcelas de bar apiladas y agrupadas

```
barplot(counts,
     main="Stacked Bar Plot",
     xlab="Treatment", ylab="Frequency",
     col=c("red", "yellow","green"),
     legend=rownames(counts))
barplot(counts,
     main="Grouped Bar Plot",
     xlab="Treatment", ylab="Frequency",
     col=c("red", "yellow", "green"),
     legend=rownames(counts), beside=TRUE)
```

Stacked bar plot

Grouped bar plot

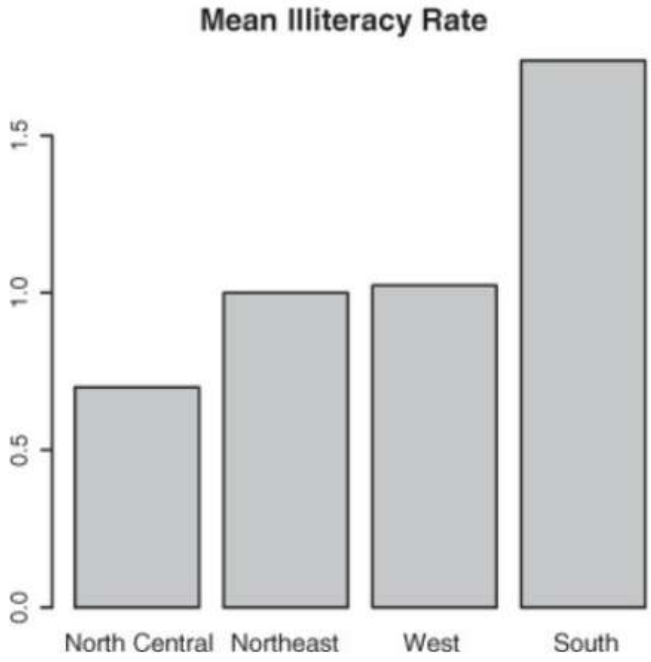
La primera función *barplot()* produce una gráfica de barras apiladas, mientras que la segunda produce una gráfica de barras agrupadas. También hemos añadido la opción col para añadir color a las barras trazadas. La leyenda. El parámetro Text proporciona etiquetas de barra para la leyenda (que solo son útiles cuando la altura es una matriz).

En el capítulo 3, cubrimos formas de formatear y colocar la leyenda para obtener el máximo beneficio. Vea si puede reorganizar la leyenda para evitar la superposición con las barras.

6.1.3. Gráficos de barras medios

Los gráficos de barras no necesitan basarse en recuentos o frecuencias. Puede crear gráficos de barras que representen medias, medianas, desviaciones estándar, etc. utilizando la función de agregado y pasando los resultados a la función barplot(). La siguiente lista muestra un ejemplo, que se muestra en la figura 6.3.

Figura 6.3. Gráfico de barras de las tasas medias de analfabetismo para las regiones de EE.UU. ordenadas por tasa



Listado 6.3. Gráfico de barras para valores medios ordenados

```
> states <- data.frame(state.region, state.x77)
> means <- aggregate(states$Illiteracy, by=list(state.region), FUN=mean)
> means
       Group.1    x
1    Northeast 1.00
2        South 1.74
3 North Central 0.70
4         West 1.02
> means <- means[order(means$x),]
> means
       Group.1    x
3 North Central 0.70
1    Northeast 1.00
4         West 1.02
2        South 1.74
> barplot(means$x, names.arg=means$Group.1)
> title("Mean Illiteracy Rate")
```

1 Sorts means, smallest to largest

2 Adds title

El listado 6.3 ordena los medios de menor a menor grandest (1). También tenga en cuenta que el uso de el título() función (2) es equivalente a agregar la opción principal en la llamada a la gráfica. medias$x es el vector que contiene las alturas de las barras, y el opción Nombres. *Arg=means$Grupo*. 1 se agrega para proporcionar Etiquetas.

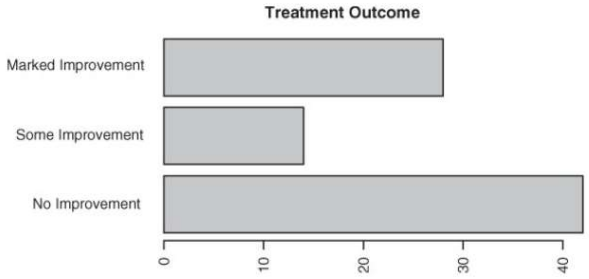
Puede llevar este ejemplo más allá. Las barras se pueden conectar con segmentos en línea recta utilizando la función lines(). También puede crear gráficos de barras medias

con intervalos de confianza superpuestos mediante la función barplot2() del paquete gplots. Consulte help(barplot2) para ver ejemplos.

6.1.4. Ajuste de gráficos de barras

Hay varias formas de ajustar la apariencia de un diagrama de bar. Por ejemplo, con muchas barras, las etiquetas de barras pueden comenzar a superponerse. Puede disminuir el tamaño de fuente mediante la opción cex.names. La especificación de valores menores que 1 reducirá el tamaño de las etiquetas. Opcionalmente, el argumento *names.arg* permite especificar un vector de caracteres de nombres utilizado para etiquetar las barras. También puede utilizar parámetros gráficos para ayudar al espaciado del texto. En el siguiente listado se da un ejemplo, con el resultado mostrado en la figura 6.4.

Figura 6.4. Gráfico de barras horizontal con etiquetas ajustadas



Listado 6.4. Colocación de etiquetas en un diagrama de barras

```
par(mar=c(5,8,4,2))
par(las=2)
counts <- table(Arthritis$Improved)
barplot(counts,
        main="Treatment Outcome",
        horiz=TRUE,
        cex.names=0.8,
      names.arg=c("No Improvement", "Some Improvement",
                  "Marked Improvement"))
```

Increases the size of the y margin

Rotates the FL bar labels

Decreases the font size in order to fit the labels comfortably

Changes the label text

La función par() le permite realizar modificaciones extensas en los gráficos que R produce de forma predeterminada. Consulte el capítulo 3 para obtener más detalles.

6.1.5. Espinogramas

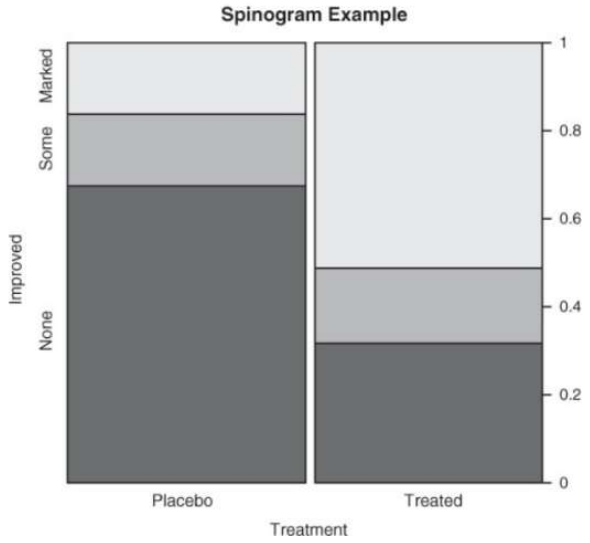
Antes de terminar nuestra discusión sobre los gráficos de barras, echemos un vistazo a una versión especializada llamada spinograma. En un spinograma, se reescala una gráfica de barras apiladas para que la altura de cada barra sea 1 y las alturas de los segmentos representen proporciones. Los espinogramas se crean a través de la función spine() del paquete vcd. El siguiente código produce un espinograma simple:

```
library(vcd)
```

```
attach(Arthritis)
counts <- table(Treatment, Improved)
spine(counts, main="Spinogram Example")
detach(Arthritis)
```

El resultado se proporciona en la figura 6.5. El mayor porcentaje de pacientes con una mejoría marcada en la condición tratada es bastante evidente en comparación con la condición de placebo.

Figura 6.5. Resultado del tratamiento del espinograma de la artritis



Además de los gráficos de barras, los gráficos circulares son un vehículo popular para mostrar la distribución de una variable categórica. Los consideraremos a continuación.

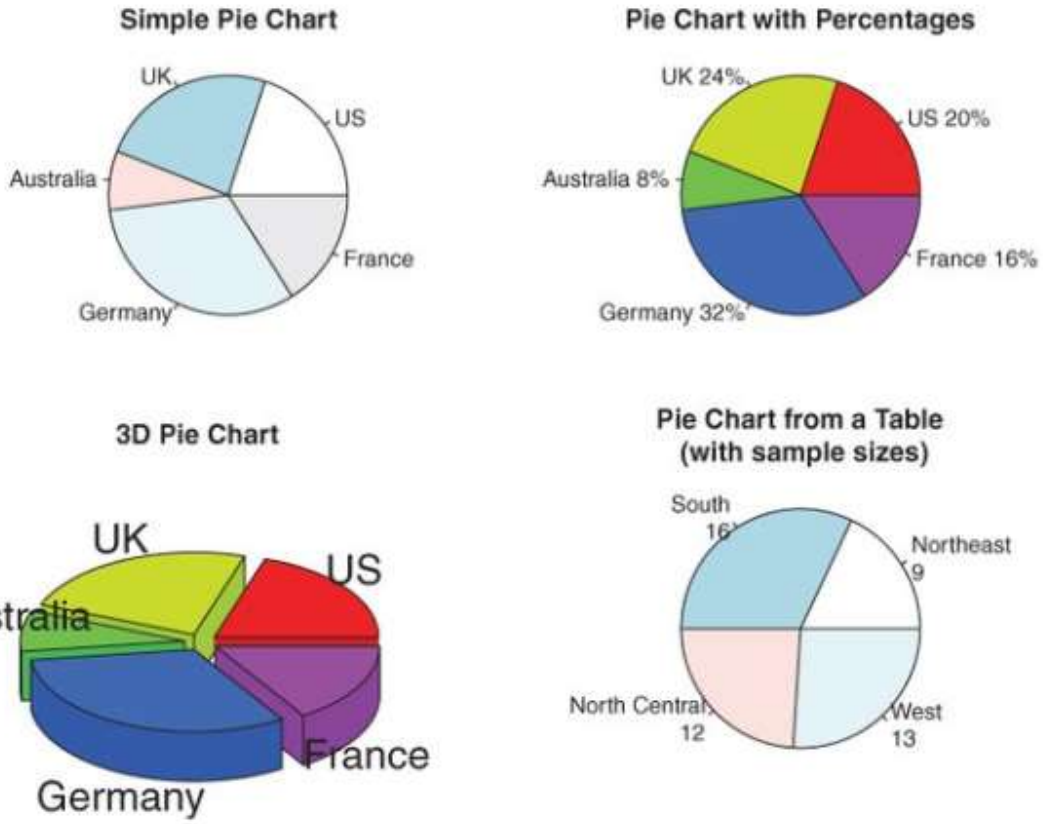
6.2. Gráficos circulares

Mientras que los gráficos circulares son omnipresentes en el mundo de los negocios, son denigrados por la mayoría de los estadísticos, incluidos los autores de la documentación de R. Recomiendan gráficos de barras o puntos sobre gráficos circulares porque las personas pueden juzgar la longitud con mayor precisión que el volumen. Quizás por esta razón, las opciones de gráfico circular en R son limitadas en comparación con otro software estadístico. Los gráficos circulares se crean con la función

```
par
```

donde x es un vector numérico no negativo que indica el área de cada sector y etiquetas proporciona un vector de caracteres de las etiquetas de sector. En la siguiente lista se dan cuatro ejemplos; las gráficas resultantes se proporcionan en la figura 6.6.

Figura 6.6. Ejemplos de gráficos circulares



Listado 6.5. Gráficos circulares

```
par(mfrow=c(2, 2))
slices <- c(10, 12,4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pie(slices, labels = lbls,
    main="Simple Pie Chart")

pct <- round(slices/sum(slices)*100)
lbls2 <- paste(lbls, " ", pct, "%", sep="")
pie(slices, labels=lbls2, col=rainbow(length(lbls2)),
    main="Pie Chart with Percentages")
library(plotrix)
pie3D(slices, labels=lbls,explode=0.1,
      main="3D Pie Chart ")
mytable <- table(state.region)
lbls3 <- paste(names(mytable), "\n", mytable, sep="")
pie(mytable, labels = lbls3,
    main="Pie Chart from a Table\n (with sample sizes)")
```

1 Combines four graphs into one

2 Adds percentages to the pie chart

3 Creates a chart from the table

Primero configura la gráfica para que se combinen cuatro gráficos en uno. (La combinación de múltiples gráficos se trata en el capítulo 3.) Luego ingresa los datos que se utilizarán para los primeros tres gráficos.

Para el segundo gráfico circular, convierta los tamaños de muestra en porcentajes y agregue la información a las etiquetas de sector. El segundo gráfico circular también define los colores de las divisiones utilizando la función ***rainbow(),*** descrita en el capítulo 3.

Aquí ***rainbow(length(lbls2))*** se resuelve en rainbow(5), proporcionando cinco colores para el gráfico.

El tercer gráfico circular es un gráfico 3D creado utilizando la función ***pie3D()*** del paquete ***plotrix***. Asegúrese de descargar e instalar este paquete antes de usarlo por primera vez. Si a los estadísticos no les gustan los gráficos circulares, desprecian positivamente los gráficos circulares 3D (aunque secretamente pueden encontrarlos bonitos). Esto se debe a que el 3Deffect no agrega información adicional sobre los datos y se considera un caramelo para distraer la vista.

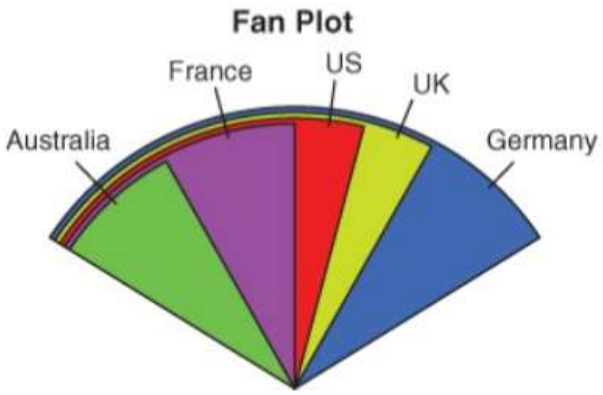
El cuarto gráfico circular muestra cómo crear un gráfico a partir de una tabla. En este caso, se cuenta el número de estados por región de EE. UU. y se agrega la información a las etiquetas antes de producir la gráfica.

Los gráficos circulares dificultan la comparación de los valores de los sectores (a menos que los valores se adjunten a las etiquetas). Por ejemplo, mirando el gráfico circular simple, ¿puede decir cómo se compara Estados Unidos con Alemania? (Si puedes, eres más perceptivo que yo). En un intento de mejorar esta situación, se ha desarrollado una variación del gráfico circular, llamada diagrama de ventilador. La trama del ventilador (Lemon & Tyagi, 2009) le proporciona una forma de mostrar tanto las cantidades relativas como las diferencias. En R, se implementa a través de la función ***fan.plot()*** en el paquete ***plotrix***.

Considere el siguiente código y el gráfico resultante (figura 6.7):

```
library(plotrix)
slices <- c(10, 12,4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
fan.plot(slices, labels = lbls, main="Fan Plot")
```

Figura 6.7. Una gráfica de fans de los datos del país



En una gráfica de ventilador, las rodajas se reorganizan para superponerse entre sí, y los radios se modifican para que cada rebanada sea visible. Aquí puede ver que Alemania es la porción más grande y que la porción de los Estados Unidos es aproximadamente un 60% más grande. Francia parece ser tan grande como Alemania y dos veces más grande que Australia. Recuerde que el ancho de la rebanada y no el radio es lo importante aquí. Como puede ver, es mucho más fácil determinar los tamaños relativos de la rebanada en un gráfico de ventilador que en un gráfico circular. Las tramas de los fans aún no se han popularizado, pero son nuevas.
Ahora que hemos cubierto los gráficos circulares y de fans, pasemos a los histogramas. A diferencia de los gráficos de barras y los gráficos circulares, los histogramas describen la distribución de una variable continua.

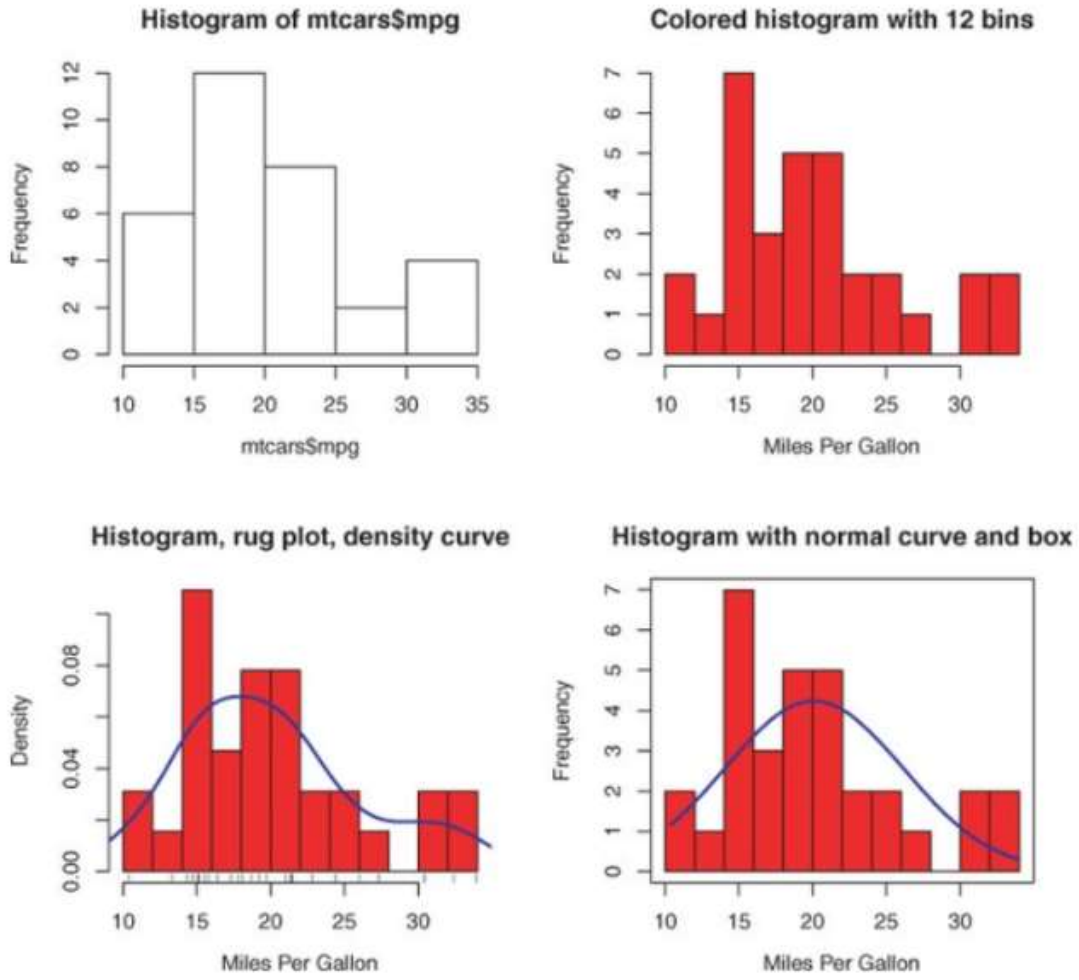
6.3. Histogramas

Los histogramas muestran la distribución de una variable continúa dividiendo el rango de puntuaciones en un número específico de contenedores en el eje x y mostrando la frecuencia de las puntuaciones en cada contenedor en el eje y. Puede crear histogramas con la función

```
hist(x)
```

donde x es un vector numérico de valores. La opción ***freq=FALSE*** crea una gráfica basada en densidades de probabilidad en lugar de frecuencias. La opción de saltos controla el número de contenedores. El valor predeterminado produce roturas igualmente espaciadas al definir las celdas del histograma. La siguiente lista proporciona el código para cuatro variaciones de un histograma; los resultados se representan en la figura 6.8.

Figura 6.8. Ejemplos de histogramas



Listado 6.6. Histogramas

```
par(mfrow=c(2,2))

hist(mtcars$mpg)                                          ❶ Simple histogram

hist(mtcars$mpg,
     breaks=12,
     col="red",
     xlab="Miles Per Gallon",
     main="Colored histogram with 12 bins")               ❷ With specified bins and color

hist(mtcars$mpg,
     freq=FALSE,
     breaks=12,
     col="red",
     xlab="Miles Per Gallon",
     main="Histogram, rug plot, density curve")           ❸ With a rug plot
rug(jitter(mtcars$mpg))
lines(density(mtcars$mpg), col="blue", lwd=2)

x <- mtcars$mpg
h<-hist(x,
        breaks=12,
        col="red",
        xlab="Miles Per Gallon",
        main="Histogram with normal curve and box")       ❹ With a normal curve and frame
xfit<-seq(min(x), max(x), length=40)
yfit<-dnorm(xfit, mean=mean(x), sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue", lwd=2)
box()
```

El primer histograma muestra la gráfica predeterminada cuando no se especifica ninguna opción. En este caso, se crean cinco contenedores y se imprimen las etiquetas y títulos de eje predeterminados. Para el segundo histograma, especificó 12 contenedores, un relleno rojo para las barras y etiquetas y títulos más atractivos e informativos. El tercer histograma mantiene los mismos colores, contenedores, etiquetas y títulos que la gráfica anterior, pero agrega una curva de densidad y una superposición de trama de alfombra. La curva de densidad es una estimación de la densidad del núcleo y se describe en la siguiente sección. Proporciona una descripción más suave de la distribución de las puntuaciones. Utilice la función lines() para superponer esta curva en un color azul y un ancho que es el doble del grosor predeterminado para las líneas. Finalmente, un gráfico de alfombra es una representación unidimensional de los valores de datos reales. Si hay muchos valores vinculados, puede ajustar los datos en el gráfico de la alfombra utilizando código como el siguiente:

```
rug(jitter(mtcars$mpag, amount=0.01))
```

Esto agrega un pequeño valor aleatorio a cada punto de datos (una variación aleatoria uniforme entre ±amount), para evitar la superposición de puntos. El cuarto histograma es similar al segundo, pero tiene una curva normal superpuesta y una caja alrededor de la figura. El código para superponer la curva normal proviene de una sugerencia publicada en la lista de correo de R-help por Peter Dalgaard. La caja circundante es producida por la función box().

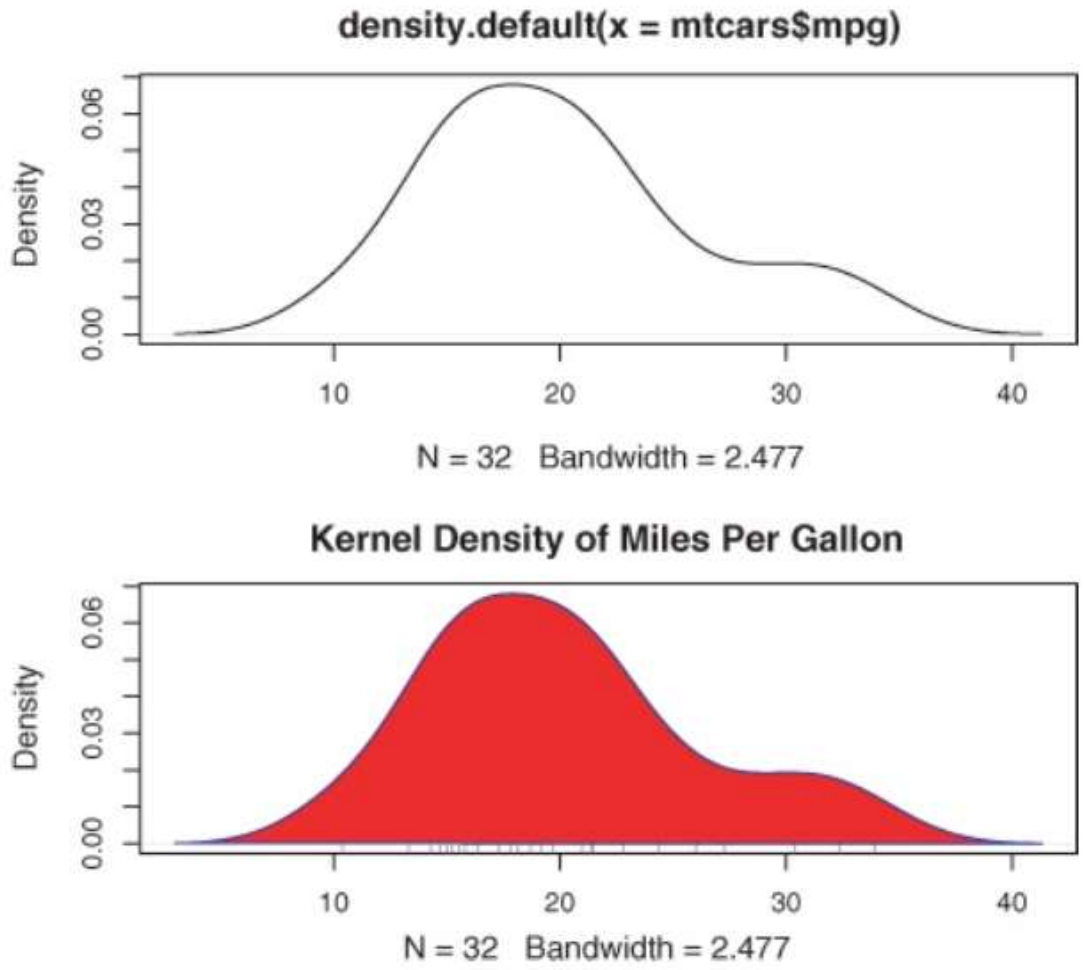
6.4. Diagramas de densidad del núcleo

En la sección anterior, vio una gráfica de densidad de kernel superpuesta a un histograma. Técnicamente, la estimación de la densidad del núcleo es un método no paramétrico para estimar la función de densidad de probabilidad de una variable aleatoria. Aunque las matemáticas están más allá del alcance de este texto, en general, las gráficas de densidad del núcleo pueden ser una forma efectiva de ver la distribución de una variable continua. El formato para un gráfico de densidad (que no se superpone en otro gráfico) es

```
plot(density(x))
```

donde x es un vector numérico. Dado que la función plot() comienza un nuevo gráfico, utilice la función lines() (listado 6.6) al superponer una curva de densidad en un gráfico existente. En la siguiente lista se dan dos ejemplos de densidad de núcleos, y los resultados se trazan en la figura 6.9.

Figura 6.9. Diagramas de densidad de kernel



Listado 6.7. Diagramas de densidad de kernel

```
par(mfrow=c(2,1))
d <- density(mtcars$mpg)

plot(d)

d <- density(mtcars$mpg)
plot(d, main="Kernel Density of Miles Per Gallon")
polygon(d, col="red", border="blue")
rug(mtcars$mpg, col="brown")
```

Creates the minimal graph with all the defaults in place

Adds a title

Adds a brown rug

Colors the curve blue and fills the area under the curve with solid red

La función polygon() dibuja un polígono cuyos vértices están dados por x y estos valores son proporcionados por la función density() en este caso. Los diagramas de densidad de kernel se pueden usar para comparar grupos. Este es un enfoque muy subutilizado, probablemente debido a una falta general de software de fácil acceso. Afortunadamente, el paquete sm llena este vacío muy bien. La función sm.density.compare() del paquete sm permite superponer los gráficos de densidad del núcleo de dos o más grupos. El formato es

```
m.density.compare(x, factor)
```

aquí x es un vector numérico y el factor es una variable de agrupación. Asegúrese de instalar el paquete sm antes de usarlo por primera vez. Un ejemplo comparando el mpg de los coches con
cuatro, seis y ocho cilindros se proporciona en la siguiente lista.

Listado 6.8. Diagramas comparativos de densidad de kernel

```
library(sm)
attach(mtcars)

cyl.f <- factor(cyl, levels= c(4,6,8),
              labels = c("4 cylinder", "6 cylinder",
                         "8 cylinder"))

sm.density.compare(mpg, cyl, xlab="Miles Per Gallon")
title(main="MPG Distribution by Car Cylinders")

colfill<-c(2:(1+length(levels(cyl.f))))
legend(locator(1), levels(cyl.f), fill=colfill)

detach(mtcars)
```

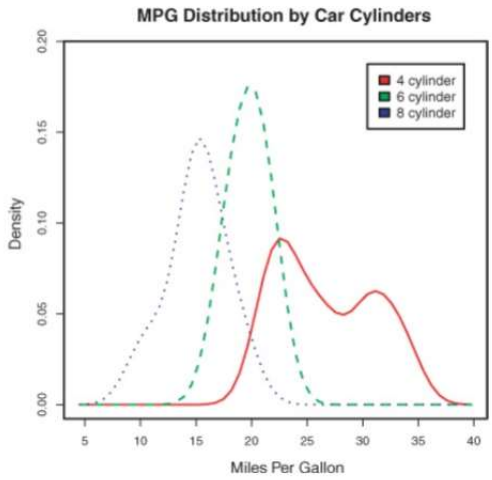
1 Creates a grouping factor

2 Plots the densities

3 Adds a legend via mouse click

Primero, se carga el paquete sm y se adjunta la trama de datos mtcars. En el marco de datos de los coches, la variable cyl es una variable numérica codificada 4, 6 u 8.cyl se transforma en un factor llamado cyl.f, con el fin de proporcionar etiquetas de valor para la gráfica. La función sm.density.compare() crea la gráfica y una instrucción title() agrega un título principal.
Por último, se añade una leyenda para mejorar la interpretabilidad. (Las leyendas están cubiertas en el capítulo 3.) Se crea un vector de colores; aquí, colfill es c(2,3,4). Luego, la leyenda se agrega a la trama a través de la función legend(). La opción locator(1) indica que colocará la leyenda de forma interactiva haciendo clic en el gráfico donde desea que aparezca la leyenda. La segunda opción proporciona un vector de caracteres de las etiquetas. La tercera opción asigna un color del colfill vectorial a cada nivel de cyl.f. Los resultados se muestran en la figura 6.10.

Figura 6.10. Diagramas de densidad del núcleo de mpg por número de cilindros



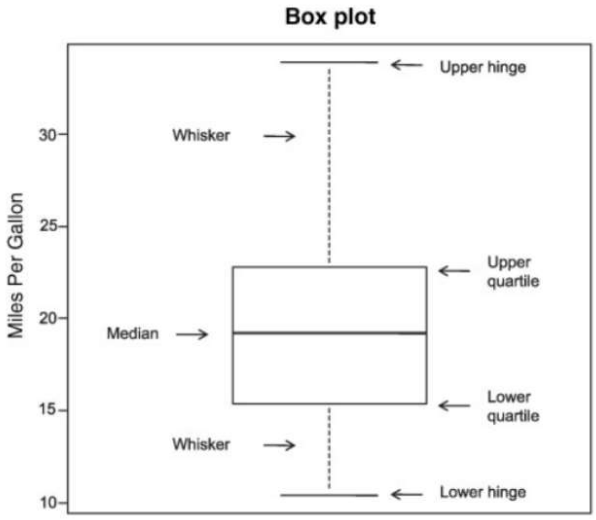
La superposición de gráficos de densidad del núcleo puede ser una forma poderosa de comparar grupos de observaciones en una variable de resultado. Aquí puede ver tanto las formas de la distribución de las puntuaciones para cada grupo como la cantidad de superposición entre los grupos. (La moraleja de la historia es que mi próximo auto tendrá cuatro cilindros, o una batería).
Los diagramas de caja también son un enfoque gráfico maravilloso (y más comúnmente utilizado) para visualizar distribuciones y diferencias entre grupos. Los discutiremos a continuación.

6.5. Diagramas de caja

Un gráfico de caja y bigotes describe la distribución de una variable continúa trazando su resumen de cinco números: el cuartil mínimo, inferior (25º percentil), la mediana (percentil 50), el cuartil superior (percentil 75) y el máximo. También puede mostrar observaciones que pueden ser valores atípicos (valores fuera del rango de ± 1.5 * IQR, donde IQR es el rango intercuartílico definido como el cuartil superior menos el cuartil inferior). Por ejemplo, esta declaración produce la gráfica que se muestra en la figura 6.11:

```
boxplot(mtcars$mpg, main="Box plot", ylab="Miles per Gallon")
```

Figura 6.11. Diagrama de caja con anotaciones añadidas a mano



Agregué anotaciones a mano para ilustrar los componentes.

De forma predeterminada, cada bigote se extiende hasta el punto de datos más extremo, que no es más de 1,5 veces el rango intercuartílico de la caja. Los valores fuera de este rango se representan como puntos (no se muestran aquí).

Por ejemplo, en la muestra de automóviles, la mediana de mpg es 19.2, el 50% de las puntuaciones caen entre 15.3 y 22.8, el valor más pequeño es 10.4 y el valor más grande es 33.9. ¿Cómo leí esto con tanta precisión a partir del gráfico? La emisión de boxplot.stats(mtcars$mpg) imprime el uso para construir el gráfico (en otras palabras, hice trampa). No parece haber ningún valor atípico, y hay un sesgo positivo leve (el bigote superior es más largo que el bigote inferior).

6.5.1. Uso de diagramas de caja paralelos para comparar grupos

Se pueden crear diagramas de caja para variables individuales o para variables por grupo. El formato es

```
boxplot(formula, data=dataframe)
```

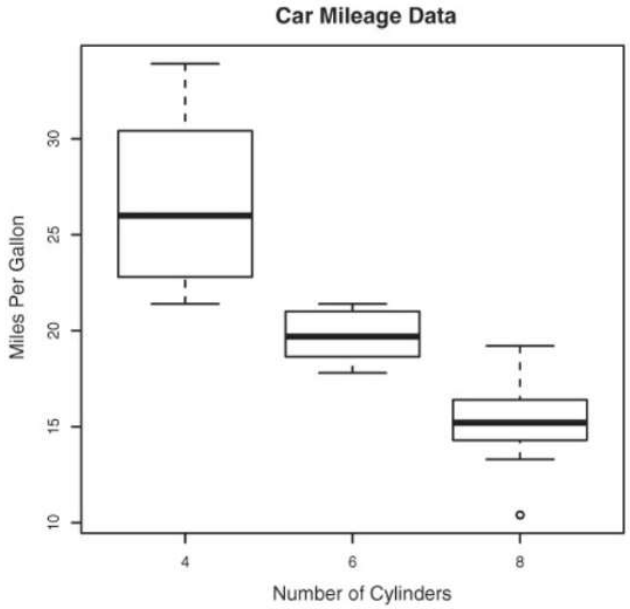
donde fórmula es una fórmula y Dataframes denota el marco de datos (o lista) que proporciona los datos. Un ejemplo de una fórmula es y ~ A, donde se genera un diagrama de caja separado para la variable numérica y para cada valor de la variable categórica A. La fórmula y ~ A*B produciría una gráfica de caja de la variable numérica y, para cada combinación de niveles en las variables categóricas A y B.

Agregar la opción Var width=TRUE hace que los anchos de diagrama de caja sean proporcionales a la raíz cuadrada de sus tamaños de muestra. Agregue horizontal=TRUE para invertir la orientación del eje.

El siguiente código revisa el impacto de cuatro, seis y ocho cilindros en autompg con gráficos de caja paralelos. La gráfica se presenta en la figura 6.12:

```
boxplot(mpg ~ cyl, data=mtcars,
        main="Car Mileage Data",
        xlab="Number of Cylinders",
        ylab="Miles Per Gallon")
```

Figura 6.12. Diagramas de caja del kilometraje del automóvil frente al número de cilindros



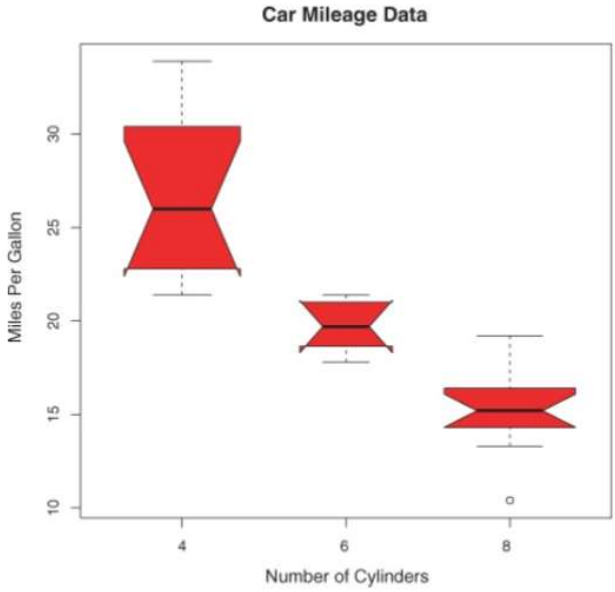
En la figura 6.12 se puede ver que hay una buena separación de grupos en función del kilometraje de gasolina. También puede ver que la distribución de mpg para automóviles de seis cilindros es más simétrica que para los otros dos tipos de automóviles. Los autos con cuatro cilindros muestran la mayor propagación (y sesgo positivo) de las puntuaciones de mpg, en comparación con los autos de seis y ocho cilindros. También hay un valor atípico en el grupo de ocho cilindros. Las parcelas de caja son muy versátiles. Al agregar notch=TRUE, obtienes gráficos de caja con muescas. Si las muescas de dos cajas no se superponen, hay una fuerte evidencia de que sus medianas difieren (Chambers et al., 1983, p. 62). El código siguiente crea diagramas de caja con muescas para el ejemplo mpg:

```
boxplot(mpg ~ cyl, data=mtcars,
        notch=TRUE,
        varwidth=TRUE,
        col="red",
        main="Car Mileage Data",
        xlab="Number of Cylinders",
        ylab="Miles Per Gallon")
```

La opción col rellena los gráficos de cuadro con un color rojo, y Var width=TRUE produce gráficos de cuadro con anchos que son proporcionales a sus tamaños de muestra.

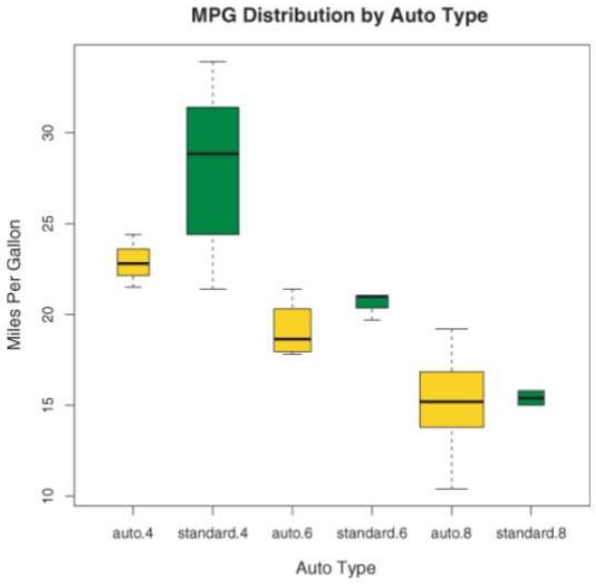
Puede ver en la figura 6.13 que el kilometraje medio del automóvil para los automóviles de cuatro, seis y ocho cilindros difiere. El kilometraje disminuye claramente con el número de cilindros.

Figura 6.13. Diagramas de caja con muescas para el kilometraje del automóvil frente al número de cilindros



Finalmente, puede producir diagramas de caja para más de un factor de agrupación. La lista 6.9 proporciona gráficos de caja para mpg frente al número de cilindros y el tipo de transmisión en un automóvil (véase la figura 6.14). Una vez más, utiliza la opción col para rellenar los diagramas de caja con color. Tenga en cuenta que los colores se reciclan; en este caso, hay seis gráficos de caja y solo dos colores especificados, por lo que los colores se repiten tres veces.

Figura 6.14. Diagramas de caja para el kilometraje del automóvil frente al tipo de transmisión y el número de cilindros



Listado 6.9. Diagramas de caja para dos factores cruzados

```
mtcars$cyl.f <- factor(mtcars$cyl,
                       levels=c(4,6,8),
                       labels=c("4","6","8"))

mtcars$am.f <- factor(mtcars$am,
                      levels=c(0,1),
                      labels=c("auto", "standard"))

boxplot(mpg ~ am.f *cyl.f,
        data=mtcars,
        varwidth=TRUE,
        col=c("gold","darkgreen"),
        main="MPG Distribution by Auto Type",
        xlab="Auto Type", ylab="Miles Per Gallon")
```

Creates a factor for the number of cylinders

Creates a factor for transmission type

Generates the box plot

De la figura 6.14, nuevamente está claro que el kilometraje medio disminuye con el número de cilindros. Para los automóviles de cuatro y seis cilindros, el kilometraje es más alto en las transmisiones estándar. Pero para los autos de ocho cilindros, no parece haber una diferencia. También puede ver en los anchos de los diagramas de caja que los autos estándar de cuatro cilindros y los autos automáticos de ocho cilindros son los más comunes en este conjunto de datos.

6.5.2. Tramas para violín

Antes de terminar nuestra discusión sobre las tramas de caja, vale la pena examinar la variación llamada trama de violín. Una trama de violín es una combinación de una trama de caja y una gráfica de densidad de núcleo. Puede crear uno utilizando la función vioplot() del paquete vioplot. Asegúrese de instalar el paquete vioplot antes del primer uso.

El formato de la función vioplot() es

```
vioplot(x1, x2, ... , names=, col=)
```

donde x1, x2, ... representan uno o más vectores numéricos a trazar (se produce una gráfica de violín para cada vector). El parámetro names proporciona un vector de caracteres de etiquetas para las gráficas de violín, y col es un vector que especifica los colores para cada gráfica de violín. Un ejemplo se da en la siguiente lista.

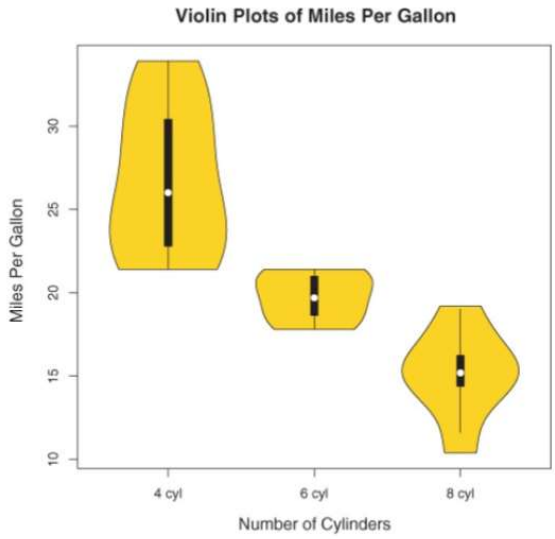
Listado 6.10. Tramas de violín

```
library(vioplot)
x1 <- mtcars$mpg[mtcars$cyl==4]
x2 <- mtcars$mpg[mtcars$cyl==6]
x3 <- mtcars$mpg[mtcars$cyl==8]
vioplot(x1, x2, x3,
        names=c("4 cyl", "6 cyl", "8 cyl"),
        col="gold")
title("Violin Plots of Miles Per Gallon", ylab="Miles Per Gallon",
```

```
        xlab="Number of Cylinders"
)
```

Tenga en cuenta que la función vioplot() requiere que separe los grupos que se van a trazar en variables separadas. Los resultados se muestran en la figura 6.15.

Figura 6.15. Gráficos de violín de mpg vs. número de cilindros



Las gráficas de violín son básicamente gráficas de densidad de núcleo superpuestas en forma de imagen especular sobre gráficas de caja. Aquí, el punto blanco es la mediana, las cajas negras van desde el cuartil inferior al superior, y las delgadas líneas negras representan los bigotes. La forma externa proporciona la gráfica de densidad del núcleo. Las tramas de violín aún no se han popularizado. Una vez más, esto puede deberse a la falta de software de fácil acceso; el tiempo lo dirá. Terminaremos este capítulo con un vistazo a los diagramas de puntos. A diferencia de los gráficos que has visto anteriormente, los diagramas de puntos trazan todos los valores de una variable.

6.6. Diagramas de puntos

Los diagramas de puntos proporcionan un método para trazar un gran número de valores etiquetados en una escala horizontal simple. Los crea con la función dotchart(), utilizando el formato

```
dotchart(x, labels=)
```

donde x es un vector numérico y etiquetas especifica un vector que etiqueta cada punto. Puede agregar una opción de grupos para designar un factor que especifique cómo se agrupan los elementos de x. Si es así, la opción gcolor controla el color de la etiqueta de grupos y cex controla el tamaño de las etiquetas. Aquí hay un ejemplo con el conjunto de datos mtcars:

```
dotchart(mtcars$mpg, labels=row.names(mtcars), cex=.7,
         main="Gas Mileage for Car Models",
```

```
      xlab="Miles Per Gallon")
```

La gráfica resultante se da en la figura 6.16. Este gráfico le permite ver el mpg para cada marca de automóvil en el mismo eje horizontal. Los diagramas de puntos generalmente se vuelven más interesantes cuando se ordenan y los factores de agrupación se distinguen por símbolo y color. En el siguiente listado se da un ejemplo que se muestra en la figura 6.17.

Figura 6.16. Diagrama de puntos de mpg para cada modelo de automóvil

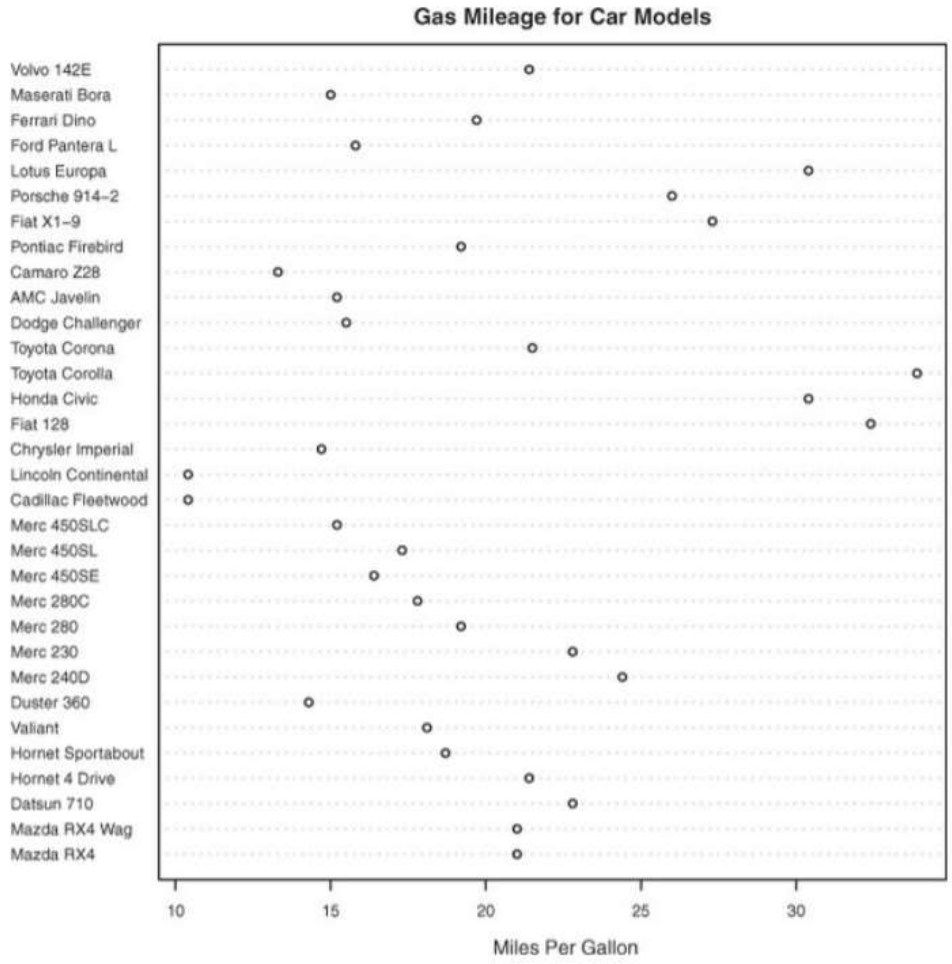
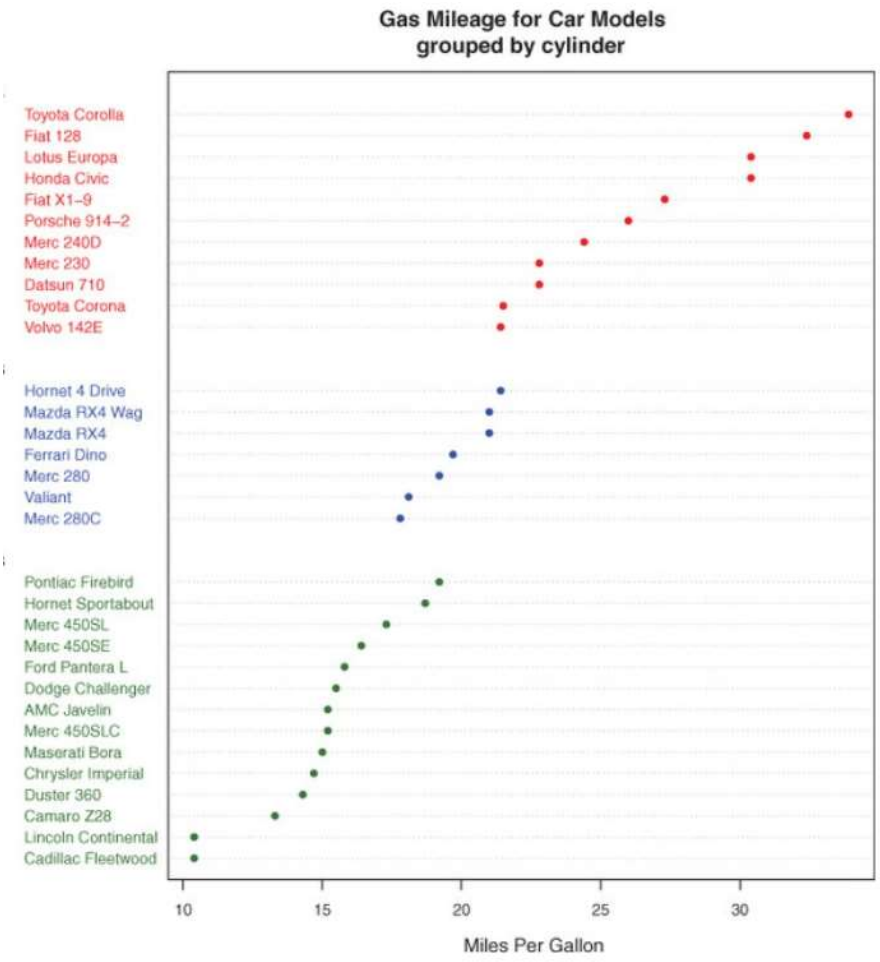


Figura 6.17. Diagrama de puntos de mpg para modelos de automóviles agrupados por número de cilindros



Listado 6.11. Diagrama de puntos agrupado, ordenado y coloreado

```
x <- mtcars[order(mtcars$mpg),]
x$cyl <- factor(x$cyl)

x$color[x$cyl==4] <- "red"
x$color[x$cyl==6] <- "blue"
x$color[x$cyl==8] <- "darkgreen"

dotchart(x$mpg,
         labels = row.names(x),
         cex=.7,
         groups = x$cyl,
         gcolor = "black",
         color = x$color,
         pch=19,
         main = "Gas Mileage for Car Models\ngrouped by cylinder",
         xlab = "Miles Per Gallon")
```

Transforms the numeric vector cyl into a factor

Sorts the data frame mtcars by mpg (lowest to highest) and saves it as data frame x

Adds a character vector (color) to data frame x containing the value "red", "blue", or "darkgreen" depending on the value of cyl

Prints the numbers 4, 6, and 8 in black

The colors of the points and labels are derived from the color vector.

Groups data points by number of cylinders

The labels for the data points are taken from the row names of the data frame (car makes).

En la figura 6.17, una serie de características se hacen evidentes por primera vez. Una vez más, se ve un aumento en el kilometraje de gasolina a medida que disminuye el número de cilindros. Pero también se ven excepciones. Por ejemplo, el Pontiac Firebird, con ocho

cilindros, obtiene un mayor kilometraje de gasolina que el Mercury 280C y el Valiant, cada uno con seis cilindros. El Hornet 4 Drive, con seis cilindros, obtiene las mismas millas por galón que el Volvo 142E, que tiene cuatro cilindros. También está claro que el Toyota Corolla obtiene el mejor kilometraje de gasolina con diferencia, mientras que el Lincoln Continental y el Cadillac Fleetwood son valores atípicos en el extremo inferior. Puede obtener información significativa de un diagrama de puntos en este ejemplo porque cada punto está etiquetado, el valor de cada punto es inherentemente significativo y los puntos están dispuestos de una manera que promueve las comparaciones. Pero a medida que aumenta el número de puntos de datos, la utilidad del diagrama de puntos disminuye.

Hay muchas variaciones del diagrama de puntos. Jacoby (2006) proporciona una discusión muy informativa del diagrama de puntos e incluye código R para aplicaciones innovadoras. Además, el paquete Hmisc ofrece una función dotplot (acertadamente llamada dotchart2()) con una serie de características adicionales.

6.7. Resumen

En este capítulo, aprendió a describir variables continuas y categóricas. Viste cómo los gráficos de barras y (en menor medida) los gráficos circulares se pueden usar para obtener información sobre la distribución de una variable categórica, y cómo los gráficos de barras apilados y agrupados pueden ayudarte a comprender cómo difieren los grupos en un resultado categórico. También exploramos cómo los histogramas, gráficos, diagramas de caja, gráficos de alfombra y diagramas de puntos pueden ayudarlo a visualizar la distribución de variables continuas. Finalmente, exploramos cómo los gráficos de densidad de kernel superpuestos, los diagramas de caja paralelos y los diagramas de puntos agrupados pueden ayudarlo a visualizar las diferencias de grupo en una variable de resultado continuo. En capítulos posteriores, ampliaremos este enfoque univariado para incluir métodos gráficos bivariados y multivariados. Verá cómo representar visualmente las relaciones entre muchas variables a la vez utilizando métodos como gráficos de dispersión, gráficos de líneas multigrupo, gráficos de mosaico, correlogramos, gráficos de red y más. En el siguiente capítulo, veremos métodos estadísticos básicos para describir distribuciones y relaciones bivariadas numéricamente, así como métodos inferenciales para evaluar si existen relaciones entre variables o se deben a un error de muestreo.