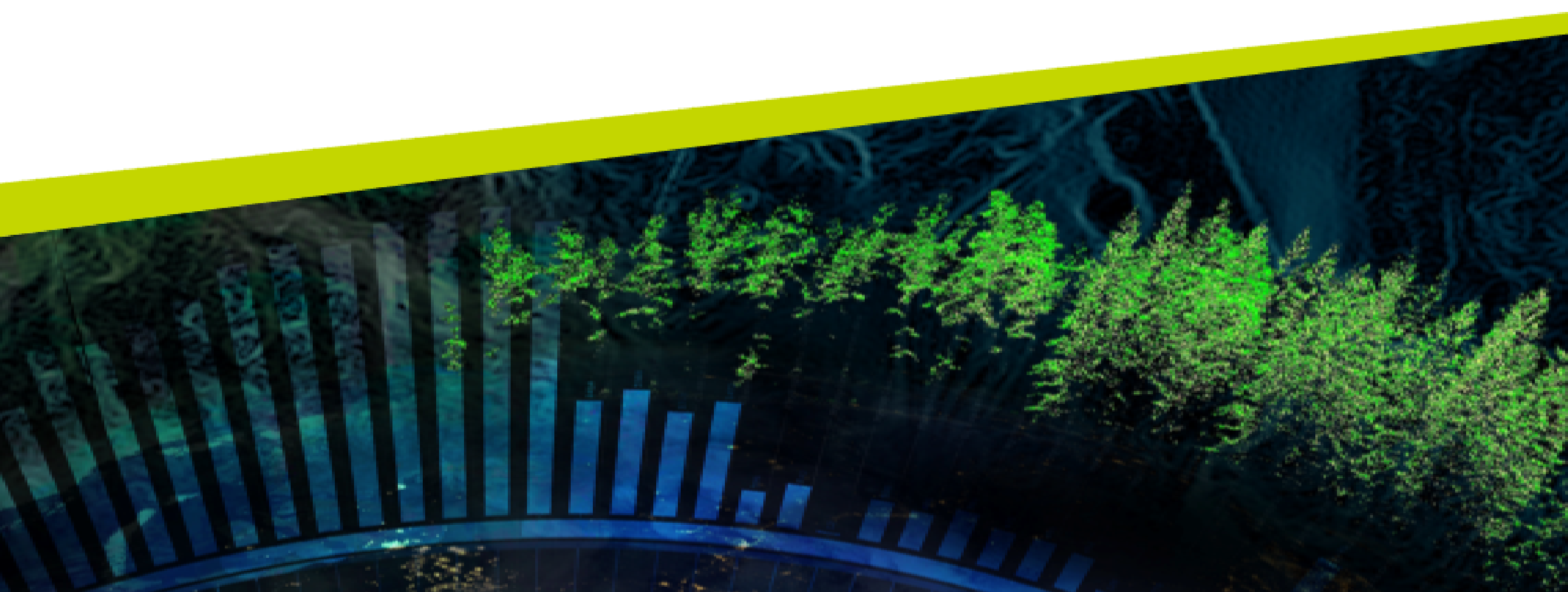
INFERENCIA Y MODELOS ESTADÍSTICOS

Jacqueline Köhler C. y José Luis Jara V.



CAPÍTULO 2. EXPLORACIÓN DE DATOS

Siempre es bueno que nos familiaricemos con los datos y algunas de sus características antes de empezar a trabajar con ellos. Esto nos ayuda a decidir qué herramientas son las más adecuadas para dar respuesta a las preguntas que queramos responder. En este capítulo revisaremos las principales estadísticas descriptivas que nos ayudarán a resumir los datos para entenderlos mejor, así como diversos tipos de gráficos que nos permitirán representar los datos de modo que podamos comprenderlos de forma visual. Para ello, tomamos como base los conceptos expuestos en Diez y col. (2017, pp. 26-50), Field y col. (2012, pp. 19-27) y STDHA (s.f.), fuentes que puedes consultar si deseas saber más acerca de estos temas.

Para muchos de los ejemplos de este capítulo usaremos el conjunto de datos `mtcars` con las modificaciones realizadas en el script 1.7, cuyo diccionario de datos se muestra en la tabla 2.1.

Variable	**Descripción**
Rendimiento	Rendimiento, en millas / galón (EEUU).
Cilindrada	Número de cilindros (4 cilindros, 6 cilindros, 8 cilindros).
Desplazamiento	Desplazamiento, en pulgadas cúbicas.
Potencia	Potencia, en caballos de fuerza brutos.
Eje	Razón del eje trasero.
Peso	Peso, en miles de libras.
Cuarto_milla	Tiempo que tarda en recorrer un cuarto de milla partiendo desde el reposo, en segundos.
Motor	Tipo de motor (V, Recto).
Transmision	Tipo de transmisión (Automático, Manual).
Cambios	Número de cambios hacia adelante (3 cambios, 4 cambios, 5 cambios).
Carburadores	Número de carburadores.

Tabla 2.1: descripción de las variables para el conjunto de datos `mtcars`.

2.1 ESTADÍSTICAS DESCRIPTIVAS

Las estadísticas descriptivas son medidas que nos permiten sintetizar y, como su nombre lo indica, describir los datos. Estas pueden aplicarse tanto a una muestra como a una población. Cuando una de estas medidas se aplica a la muestra, corresponde a un **estimador puntual** de la misma medida para la población. Al ser una estimación, no es exacta, aunque la precisión tiende a aumentar mientras mayor sea el tamaño de la muestra.

Un concepto importante a tener en cuenta es la noción de **distribución**. En este capítulo se considera la **distribución de frecuencia**, que representa cuántas veces aparece cada valor para una variable en un conjunto de datos.

2.1.1 Estadísticas descriptivas para datos numéricos

Una de las estadísticas descriptivas más empleadas es la **media**, conocida en otros contextos como media aritmética o promedio. Denotamos la **media muestral** por $\overline{x}$, donde x corresponde al nombre de la variable, mientras que para la **media poblacional** empleamos la notación μ_x. Esta medida se calcula como muestra la ecuación 2.1, donde x_i son los n valores observados de la variable. Podemos entender la media como el punto de equilibrio de la distribución (Diez y col., 2017, p. 28). Así, la media corresponde a una **medida de tendencia central**.

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2.1}$$

El script 2.1 muestra cómo usar la función `mean()` de R para calcular la media de diversas variables del conjunto de datos `mtcars`[1]. Como primer ejemplo, se calcula la media de la variable `Rendimiento`. A continuación se muestra cómo realizar esta operación para dos variables, señaladas por el índice de sus respectivas columnas. Luego, de manera similar, se calculan las medias para cuatro columnas consecutivas de la matriz de datos. En estos dos casos hacemos uso de la función `sapply()`, que permite aplicar una misma función (cualquiera) para múltiples columnas.

Script 2.1: uso de las funciones `mean()` y `sapply()`.

```r
# Cargar conjunto de datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

# Calcular la media para la variable Rendimiento.
media <- mean(datos[["Rendimiento"]])
cat("Rendimiento medio:", media, "\n\n")

# Calcular la media para la tercera y quinta columnas
# (variables Desplazamiento y Eje).
cat("Medias\n")
print(sapply(datos[c(3, 5)], mean))
cat("\n")

# Calcular la media para las columnas 3 a 6
# (variables Desplazamiento, Potencia, Eje y Peso).
cat("Medias\n")
print(sapply(datos[3:6], mean))
cat("\n")

# Calcular la media para la variable Rendimiento omitiendo valores faltantes.
print(mean(datos[["Rendimiento"]], na.rm = TRUE))
```

La función `mean()` devuelve `NA` (*not available*, es decir, no disponible) si existen valores faltantes en los datos de entrada. Para prevenir este error, se puede proporcionar un argumento adicional que descarte los valores faltantes, como muestra la última línea del script 2.1.

Una medida de tendencia central alternativa a la media es la **mediana**, que es, simplemente, el valor central de los valores previamente ordenados. Cuando no existe un valor central, vale decir, cuando el tamaño de la muestra es par, la mediana está dada por el promedio simple de los dos valores centrales. En R, la mediana se calcula con la función `median()`.

[1] Todas las demás funciones de R mencionadas en esta sección para las que no se proporcione un script se usan del mismo modo que `mean()`.

La **moda** es, simplemente, el valor más frecuente en el conjunto de datos. No obstante, tiene el problema de que puede haber múltiples modas. Dependiendo de la cantidad de modas, se habla de distribuciones **unimodales**, **bimodales** y **multimodales**.

Si bien R no cuenta con una función nativa para encontrar la moda, el paquete `modeest` ofrece la función `mfv()` que entrega el valor más frecuente de una variable. En caso de que dos (o más) valores sean los más frecuentes con igual cantidad de observaciones, los entrega todos en forma de vector.

Las medidas que hemos estudiado hasta ahora buscan describir el centro del conjunto de datos. No obstante, también es importante conocer su **variabilidad o dispersión**, pues así se puede saber qué tan semejantes (o diferentes) son las observaciones entre sí. Estas suelen calcularse en base a la **desviación** de las observaciones, que se entiende como la distancia entre una observación y la media del conjunto de datos. Las dos principales medidas de dispersión son la **varianza** y la **desviación estándar**, ambas basadas en los cuadrados de las distancias, ya que, por una parte, los valores grandes se incrementan más significativamente y, por otra, se opera solo con valores positivos, pues la dirección de la distancia no es de interés.

La varianza muestral se calcula como muestra la ecuación 2.2, donde x_i son los valores de cada una de las n observaciones. Cabe destacar que puede emplearse un subíndice para indicar el nombre de la variable, al igual que en el caso de la media.

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 \tag{2.2}$$

La desviación estándar de la muestra se define como la raíz cuadrada de la varianza (2.3), medida que resulta de gran utilidad cuando se necesita saber cuán cercanos son los datos a la media, ya que se encuentra en la misma escala que la variable.

$$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{2.3}$$

Al igual que en el caso de la media, podemos usar las fórmulas anteriores para obtener estimaciones puntuales de la varianza y la desviación estándar de la población, denotadas por σ^2 y σ, respectivamente.

Es importante considerar que, si bien la media y la desviación estándar permiten conocer el centro y la dispersión del conjunto de datos, respectivamente, la distribución de los puntos puede ser muy diferente, como ilustra la figura 2.1.

Las funciones de R para calcular la varianza y la desviación estándar son, respectivamente, `var()` y `sd()`.

Aunque menos empleado, el **rango** muestra los valores extremos, es decir, el mínimo y el máximo, de una variable. R ofrece la función `range()` para obtener ambos valores, además de `min()` y `max()` para obtenerlos por separado.

En párrafos anteriores vimos que la mediana es el valor central (o el promedio de los dos valores centrales) del conjunto de datos ordenado, ya sea una población o una muestra. Esto significa, entonces, que esta medida divide el conjunto de datos en dos mitades con igual cantidad de elementos. De manera similar, es posible dividir el conjunto de datos en segmentos más pequeños, por ejemplo en 4, 10 o 100 partes con igual cantidad de elementos. Cada fragmento del conjunto de datos dividido de esta forma recibe el nombre de **cuantil**. Algunas subdivisiones de uso frecuente reciben nombres especiales:

- **Percentiles**: dividen el conjunto de datos en 100 subconjuntos de igual tamaño.
- **Deciles**: dividen el conjunto de datos en 10 subconjuntos de igual tamaño.
- **Quintiles**: dividen el conjunto de datos en 5 subconjuntos de igual tamaño.
- **Cuartiles**: dividen el conjunto de datos en 4 subconjuntos de igual tamaño.

Los cuantiles (al igual que las otras subdivisiones antes mencionadas) se nombran de forma ascendente según el sentido de crecimiento del conjunto de datos. Así, el percentil 1 contiene a los valores más pequeños,

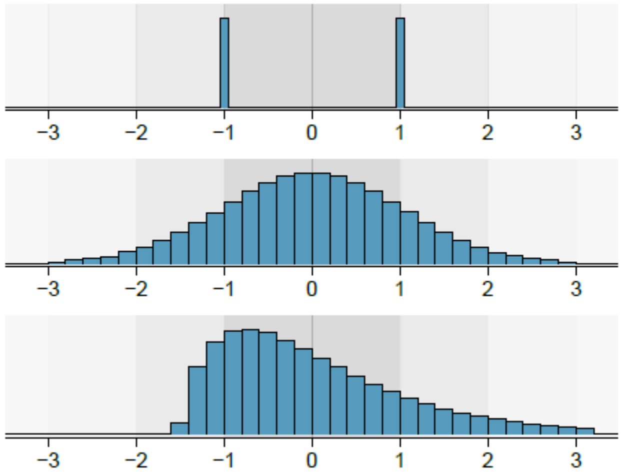


Figura 2.1: tres distribuciones de población muy distintas con media $\mu = 0$ y desviación estándar $\sigma = 1$. Fuente: Diez y col. (2017, p. 34).

mientras que el percentil 100, a los más grandes. Cabe destacar que la mediana corresponde al percentil 50 o al cuartil 2, y que nombrar al decil 3 es equivalente al percentil 30.

R proporciona la función `quantile()` para calcular cuantiles, que por defecto calcula los cuartiles, aunque su uso puede generalizarse mediante el parámetro adicional `probs`, como muestra el script 2.2. La función `seq()` genera una secuencia de números equiespaciados, y recibe como argumentos el inicio, el término y el incremento de la secuencia.

Script 2.2: cálculo de cuantiles con la función `quantile()`.

```r
# Cargar conjunto de datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

# Cálculo de percentiles para la variable Rendimiento.
cat("Cuartiles:\n")
print(quantile(datos[["Rendimiento"]]))
cat("\n")

cat("Quintiles:\n")
print(quantile(datos[["Rendimiento"]], seq(0, 1, 0.2)))
cat("\n")

cat("Deciles:\n")
print(quantile(datos[["Rendimiento"]], seq(0, 1, 0.1)))
cat("\n")

cat("Percentiles:\n")
print(quantile(datos[["Rendimiento"]], seq(0, 1, 0.01)))
```

Ahora que conocemos los cuartiles, podemos introducir una nueva medida de variabilidad que usaremos a menudo, llamada **rango intercuartil** o IQR (por su sigla en inglés), dada por la ecuación 2.4, donde Q_1 y Q_3 corresponden a los cuartiles 1 y 3, respectivamente. Al igual que la varianza y la desviación estándar, mientras más disperso sea el conjunto de datos, mayor será el valor del IQR. En R, la función que calcula este estimador es `IQR()`.

$$IQR = Q_3 - Q_1 \tag{2.4}$$

Muchas veces los conjuntos de datos contienen lo que se conoce como **valores atípicos** o ***outliers***. Estos corresponden a observaciones que parecen estar fuera de rango o ser muy extremos con respecto al resto de los datos. Medidas como la media o la desviación estándar son muy sensibles a los valores atípicos, por lo que son propensas a errores ante la presencia de este tipo de observaciones. Para reducir el efecto de los valores extremos muchas veces necesitaremos medidas **robustas**, que son aquellas que proporcionan una estimación confiable aún ante la presencia de valores atípicos. En este escenario, la mediana resulta ser una buena medida de tendencia central y el IQR, una buena medida de dispersión.

Nos encontraremos frecuentemente con la necesidad de calcular varias medidas de tendencia central y de dispersión descritas en el apartado anterior. Por esta razón, R, y algunos de sus paquetes, ofrecen algunas funciones que calculan varios de estos estadísticos con una sola llamada. Tal es el caso de la función nativa `summary()`, que entrega la media, la mediana, el primer y el tercer cuartil, el mínimo y el máximo. Otra función que nos puede ser de mucha ayuda es `summarise()`, del paquete `dplyr`. Con ella podemos calcular varias de las medidas en una sola llamada, como muestra el script 2.3.

Script 2.3: uso de la función `summarise()` del paquete `dplyr`.

```
library(dplyr)

datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

# Cálculo de varias medidas para la variable Potencia.
medidas_potencia <- datos %>% summarise(Media = mean(Potencia),
                                        Mediana = median(Potencia),
                                        Varianza = var(Potencia),
                                        IQR = IQR(Potencia))

print(medidas_potencia)
cat("\n")

# Cálculo de la media y la desviación estándar para las variables Peso y
# Cuarto_milla.
medidas_varias <- datos %>% summarise(Media_P = mean(Peso),
                                      Media_C = median(Cuarto_milla),
                                      SD_P = sd(Peso),
                                      SD_C = sd(Cuarto_milla))

print(medidas_varias)
cat("\n")
```

2.1.2 Estadísticas descriptivas para datos categóricos

Cuando queremos trabajar con datos categóricos, medidas como la media o la desviación estándar carecen de sentido. En consecuencia, necesitamos otros estadísticos para resumir el conjunto de datos.

Como primer estadístico para variables categóricas podemos mencionar la **frecuencia**, que corresponde a la cantidad de veces que podemos encontrar cada nivel de la variable en los datos. Otro estadístico importante corresponde a la **proporción**, que corresponde a la frecuencia relativa. En otras palabras, la proporción corresponde a frecuencia de un nivel de la variable dividida por la cantidad total de observaciones.

La mejor alternativa para este tipo de datos es la **tabla de contingencia**, también llamada **matriz de confusión** o **tabla de frecuencias**, donde cada fila representa la cantidad de veces en que ocurre una combinación de variables. También es posible usar porcentajes o proporciones en lugar de la cantidad de ocurrencia, en cuyo caso se habla de una **tabla de frecuencias relativas**. La tabla 2.2 muestra la tabla de contingencia (de frecuencias) para la variable `Cambios`. Se puede observar, por ejemplo, que el conjunto de datos contiene una muestra de 32 automóviles y que 15 de ellos tienen tres cambios.

3 cambios	4 cambios	5 cambios	Total
15	12	5	32

Tabla 2.2: tabla de contingencia para la cantidad de cambios de los automóviles.

Desde luego, podemos construir tablas de contingencia de manera bastante sencilla en R. El script 2.4 muestra dos formas de obtener la tabla 2.2. La primera es la función `table()` y la segunda, la función `xtabs()`. El funcionamiento de ambas es equivalente, aunque `xtabs()` muestra el nombre de la variable tabulada al imprimir los resultados y `table()` no lo hace. Las tablas entregadas por estas funciones no incluyen los totales por filas, pero la función `marginSums()` permite calcularlos y mostrarlos como un vector. A su vez, la función `addmargins()` permite calcular dichos totales e incorporarlos a la tabla. Para terminar, el las últimas sentencias del script 2.4 ilustran la manera de obtener las tablas de frecuencias relativas con proporciones y porcentajes, respectivamente.

Podemos ver que las llamadas a `table()` y a `xtabs()` son algo diferentes. La primera recibe como argumento la columna de la matriz de datos, es decir, un vector con los datos a tabular, mientras que la segunda recibe una fórmula en que no existe una variable dependiente y la variable categórica es la independiente.

Script 2.4: tabla de contingencia para la variable `Cambios`.

```r
# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

# Crear tabla de contingencia para la variable gear.
contingencia <- table(datos[["Cambios"]])
cat("Tabla de contingencia generada con table():\n")
print(contingencia)
cat("\n")

# Otra forma de crear la misma tabla.
contingencia <- xtabs(~ Cambios, data = datos)
cat("Tabla de contingencia generada con xtabs():\n")
print(contingencia)
cat("\n")

# Calcular totales por fila y mostrarlos por separado.
totales <- marginSums(contingencia)
cat("Totales por fila:\n")
print(totales)
cat("\n")

# Calcular totales por fila y agregarlos a la tabla.
con_totales <- addmargins(contingencia, 1)
cat("Tabla de contingencia con totales por fila:\n")
print(con_totales)
cat("\n")

# Convertir a tabla de proporciones
proporciones <- prop.table(contingencia)
proporciones <- addmargins(proporciones, 1)
cat("Tabla de contingencia con proporciones:\n")
print(proporciones)
cat("\n")

```

```r
# Convertir a tabla de porcentajes con 2 decimales.
porcentajes <- round(prop.table(contingencia), 4) * 100
porcentajes <- addmargins(porcentajes)
cat("Tabla de contingencia con porcentajes:\n")
print(porcentajes)
cat("\n")
```

También es podemos construir matrices de confusión para dos variables categóricas, como muestra la tabla 2.3 para las variables `Motor` y `Transmisión`.

		Cambios			
		3 cambios	4 cambios	5 cambios	Total
Transmision	Automático	15	4	0	19
	Manual	0	8	5	13
	Total	15	12	5	32

Tabla 2.3: tabla de contingencia para las variables `Motor` y `Transmisión`.

En ocasiones resulta útil determinar las proporciones por fila o por columna, que podemos obtener dividiendo el valor de una celda de la matriz por el total de su fila o columna, según corresponda. Así, el total de cada fila (o columna) es igual a 1. Puesto que las proporciones por fila y por columna no son equivalentes, debemos ser cuidadosos al escoger la más adecuada en cada caso. Las tablas 2.4 a 2.6 muestran las proporciones por fila, por columna y generales para la matriz de confusión de la tabla 2.3. La construcción en R de la tabla de contingencia y las tablas de proporciones para dos variables se muestra en el script 2.5.

		Cambios			
		3 cambios	4 cambios	5 cambios	Total
Transmision	Automático	0,7894737	0,2105263	0,0000000	1,0000000
	Manual	0,0000000	0,6153846	0,3846154	1,0000000

Tabla 2.4: tabla de proporciones con totales por fila para la tabla 2.3.

		Cambios		
		3 cambios	4 cambios	5 cambios
Transmision	Automático	1,0000000	0,3333333	0,0000000
	Manual	0,0000000	0,6666667	1,0000000
	Total	1,0000000	0,0000000	1,0000000

Tabla 2.5: tabla de proporciones con totales por columna para la tabla 2.3.

Script 2.5: tablas de contingencia y proporciones para dos variables.

```r
# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

# Crear tabla de contingencia para las variables Transmision y gear.
contingencia <- table(datos[["Transmision"]], datos[["Cambios"]])
cat("Tabla de contingencia generada con table():\n")
print(contingencia)
cat("\n")

# Otra forma de crear la misma tabla.
contingencia <- xtabs(~ Transmision + Cambios, data = datos)
cat("Tabla de contingencia generada con xtabs():\n")
print(contingencia)
cat("\n")
```

		Cambios			
		3 cambios	4 cambios	5 cambios	Total
Transmision	Automático	0,46875	0,12500	0,00000	0,59375
	Manual	0,00000	0,25000	0,15625	0,40625
	Total	0,46875	0,37500	0,15625	1,00000

Tabla 2.6: tabla de proporciones con totales por fila y columna para la tabla 2.3.

```

# Proporciones con totales por fila.
proporciones_fila <- prop.table(contingencia, margin=1)
proporciones_fila <- addmargins(proporciones_fila, margin=2)
cat("Tabla de contingencia con proporciones totales por fila:\n")
print(proporciones_fila)
cat("\n")

# Proporciones con totales por columna.
proporciones_columna <- prop.table(contingencia, margin=2)
proporciones_columna <- addmargins(proporciones_columna, margin=1)
cat("Tabla de contingencia con proporciones totales por columna:\n")
print(proporciones_columna)
cat("\n")

# Proporciones con totales.
proporciones <- prop.table(contingencia)
proporciones <- addmargins(proporciones)
cat("Tabla de contingencia con proporciones totales:\n")
print(proporciones)
cat("\n")
```

Aunque no ocurre con frecuencia, podríamos necesitar una matriz de confusión para más de dos variables. Veamos ahora un ejemplo con tres variables: `Motor`, `Cambios` y `Transmisión`. Para ello, tomamos una de las variables (en este caso, `Motor`) y creamos una subtabla por cada uno de sus niveles. Cada subtabla muestra las frecuencias para la combinación de las dos variables restantes cuando `Motor` tiene el nivel correspondiente, como muestra la tabla 2.7. En R, podemos obtener estas tablas como muestra el script 2.6. Desde luego, esta misma idea puede extenderse para cuatro o más variables categóricas.

Motor = Recto

		Cambios		
		3 cambios	4 cambios	5 cambios
Transmision	Automático	3	4	0
	Manual	0	6	1

Motor = V

		Cambios		
		3 cambios	4 cambios	5 cambios
Transmision	Automático	12	0	0
	Manual	0	2	4

Tabla 2.7: tabla de contingencia para tres variables.

Script 2.6: matriz de confusión para tres variables.

```
# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

```

```
# Convertir la variable Cambios en categórica.
datos[["Cambios"]] <- factor(datos[["Cambios"]])

# Crear tabla de contingencia para las variables Transmision,
# Cambios y Motor.
contingencia <- ftable(datos[["Transmision"]], datos[["Cambios"]],
                       datos[["Motor"]])

cat("Tabla de contingencia generada con ftable():\n")
print(contingencia)
cat("\n")

# Otra forma de crear la misma tabla.
xtabs(~ Cambios + Transmision + Motor, data = datos)
cat("Tabla de contingencia generada con xtabs():\n")
print(contingencia)
cat("\n")
```

2.1.3 Trabajando con datos agrupados

A menudo nos veremos en la necesidad de obtener estadísticas descriptivas de una variable separando las observaciones en grupos de acuerdo a una variable categórica. Para ello, el paquete `dplyr` ofrece la función `group_by()`, que podemos usar en conjunto con `summarise()`, como muestra el script 2.7. En dicho script, primero se agrupan las observaciones de acuerdo a la variable `Cambios`, y luego se efectúa una llamada a `summarise()` donde el primer argumento cuenta la cantidad de observaciones en el grupo actual y los argumentos restantes (que pueden ser tantos como se desee) corresponden a diferentes estadísticas descriptivas.

Script 2.7: estadísticas descriptivas para datos agrupados.

```
library(dplyr)

# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

resumen <- group_by(datos, Cambios) %>%
  summarise(count = n(), mean(Rendimiento), median(Rendimiento),
            sd(Rendimiento), IQR(Rendimiento), mean(Potencia))

print(resumen)
```

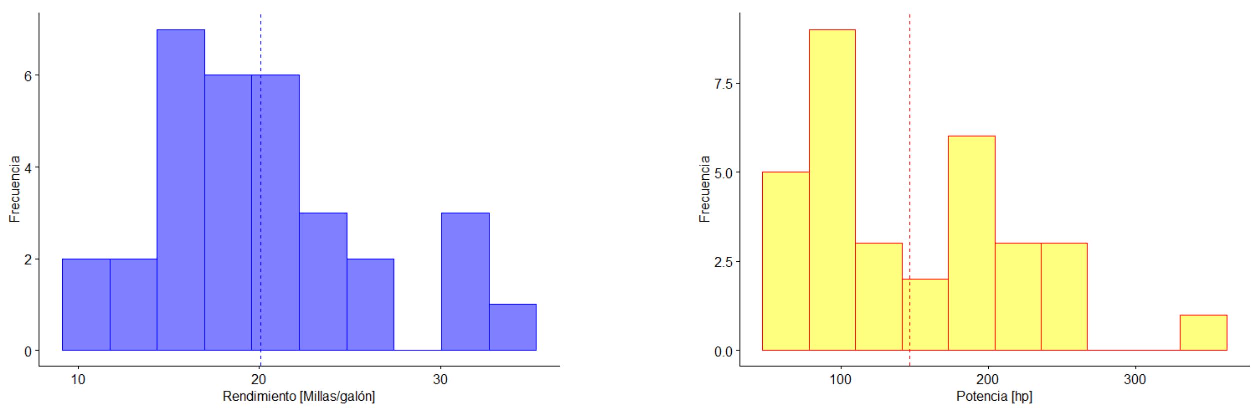
2.2 REPRESENTACIÓN GRÁFICA DE DATOS

En esta sección revisaremos diversos tipos de gráficos que resultan útiles al momento de estudiar un conjunto de datos disponibles, considerando su definición, su utilidad y cómo se construyen en R. Para crear gráficos en R usaremos el paquete `ggpubr`. Algunos de los principales parámetros que usaremos para crear y editar gráficos con este paquete son:

- `data`: un data frame.
- `x`: string con el nombre de la variable `x`.
- `y`: string(s) con el(los) nombre(s) de la(s) variable(s) a graficar.
- `color`: color de delineado.
- `fill`: color de relleno.
- `palette`: paleta de colores cuando existen múltiples grupos.
- `linetype`: tipo de línea a emplear.
- `add`: permite agregar elementos adicionales al gráfico, como barras de error o la media, entre otros.
- `title`: título del gráfico.
- `xlab`: rótulo del eje x. Puede ocultarse usando `xlab = FALSE`.
- `xlab`: rótulo del eje y. Puede ocultarse usando `ylab = FALSE`.

2.2.1 Una variable numérica

El **histograma** resulta muy útil si queremos representar una única variable numérica y la muestra es grande. Podemos decir que este gráfico muestra una aproximación a la **densidad** (o distribución de frecuencias) para la variable, para lo que tenemos que dividir el rango de valores posibles en intervalos (generalmente iguales) y luego contar la cantidad de observaciones en cada intervalo. Para construir el gráfico, creamos una barra por cada intervalo, cuya altura (o longitud) es proporcional a la cantidad de observaciones en el intervalo representado. La figura 2.2 muestra histogramas creados con el script 2.8.



(a) Distribución cercana a la simétrica.

(b) Distribución desviada a la izquierda.

Figura 2.2: dos histogramas.

Script 2.8: histogramas para las variables `Rendimiento` y `Potencia`.

```
library(ggpubr)

# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

# Histograma para la variable Rendimiento.
g1 <- gghistogram(datos,
                  x = "Rendimiento",
                  bins = 10,
                  add = "mean",
```

```
                  xlab = "Rendimiento [Millas/galón]",
                  ylab = "Frecuencia",
                  color = "blue",
                  fill = "blue")

print(g1)

# Histograma para la variable Potencia.
g2 <- gghistogram(datos,
                  x = "Potencia",
                  bins = 10,
                  add = "mean",
                  xlab = "Potencia [hp]",
                  ylab = "Frecuencia",
                  color = "red",
                  fill = "yellow")

print(g2)
```

A medida que avancemos en este libro, veremos que es muy importante conocer la **distribución de frecuencias** de una variable. Al observar la figura 2.2b, podemos ver que la frecuencia es mayor para potencias más bajas, pues las barras de la izquierda del gráfico son, en general, algo más altas que las de la derecha. Podría decirse que las observaciones se concentran a la izquierda y que hay una cola que se prolonga hacia la derecha. Cuando esto ocurre, decimos que la la distribución está **desviada a la izquierda**, o que hay **asimetría negativa**. Análogamente, podría darse que la distribución estuviese desviada a la derecha o, equivalentemente, que presenta asimetría positiva. En el caso de la figura 2.2a, el histograma es más **simétrico**, pues las observaciones se aglomeran hacia el centro y hay colas tanto a la izquierda como a la derecha. Para ilustrar mejor la idea de la simetría, podemos revisar una vez más la figura 2.1, donde la población central es perfectamente simétrica y la inferior presenta asimetría positiva.

Otra ventaja de los histogramas es que permiten identificar modas de una variable, las cuales corresponden a barras que sean más prominentes que las de su entorno. Ambos ejemplos de la figura 2.2 son bimodales, pues tienen dos modas claramente identificables.

Otro gráfico que usaremos a menudo es el de **gráfico de caja**. Es muy útil, pues su construcción considera 5 estadísticos para representar el conjunto de datos y además facilita la identificación de datos atípicos. La figura 2.3 muestra este gráfico para la variable `Potencia`, el cual fue creado con el script 2.9.

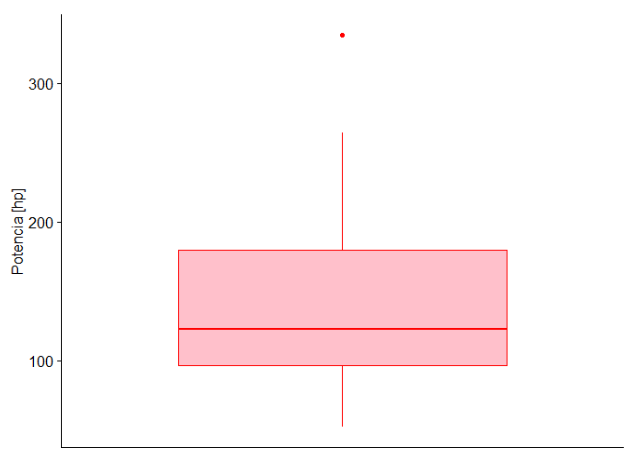


Figura 2.3: gráfico de caja para la variable `Potencia`.

Los extremos inferior y superior del rectángulo o caja de la figura 2.3 corresponden, respectivamente, al primer y al tercer cuartil, mientras que la línea horizontal al interior de la caja denota la mediana. Así, la caja engloba el 50 % central de los datos, y su altura corresponde al rango intercuartil. Las barras que se extienden por sobre y por debajo de la caja, llamadas bigotes, capturan aquellos datos fuera de la caja central y que estén situados a no más de 1,5 veces el IQR. Cualquier observación que esté más allá de la caja y los bigotes se representa como un punto, el cual podría tratarse de una observación atípica.

Script 2.9: gráfico de caja para la variable `Potencia`.

```
library(ggpubr)

# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

g <- ggboxplot(datos[["Potencia"]],
               color = "red",
               fill = "pink",
               ylab = "Potencia [hp]")

g <- g + rremove("x.ticks")
g <- g + rremove("x.text")
g <- g + rremove("x.title")

print(g)
```

2.2.2 Una variable categórica

Si queremos representar una única variable categórica, lo más adecuado es usar un **gráfico de barras**, pues cada barra es tan larga como la proporción de valores presentes en cada nivel de la variable. La figura 2.4 muestra el gráfico de barras correspondiente a la tabla 2.2, elaborado mediante el script 2.10.

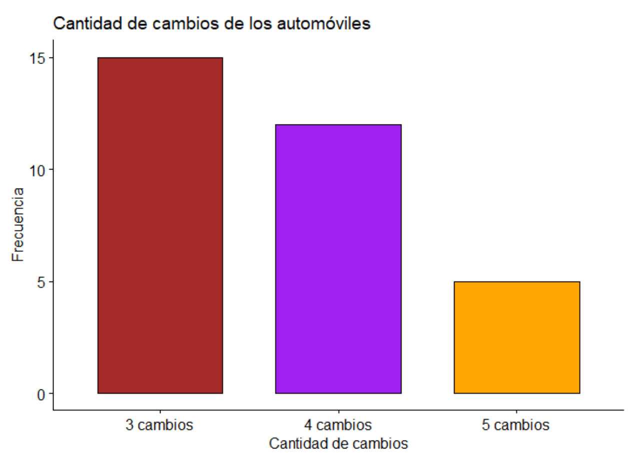


Figura 2.4: gráfico de barras para la variable `Cambios`.

Script 2.10: gráfico de barras para la variable `Cambios`.

```
library(ggpubr)

# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
```

```
                     row.names = 1)

# Crear la tabla de frecuencias para la variable Cambios y convertirla a
# data frame.
contingencia <- as.data.frame(xtabs(~ Cambios, data = datos))

# Crear el gráico de barras.
g <- ggbarplot(contingencia,
               x = "Cambios",
               y = "Freq",
               fill = c("brown", "purple", "orange"),
               title = "Cantidad de cambios de los automóviles",
               xlab = "Cantidad de cambios",
               ylab = "Frecuencia")

print(g)
```

Otra alternativa para representar una única variable categórica es el **gráfico de torta**, que se presenta en la figura 2.5 y se construye en R como muestra el script 2.11.

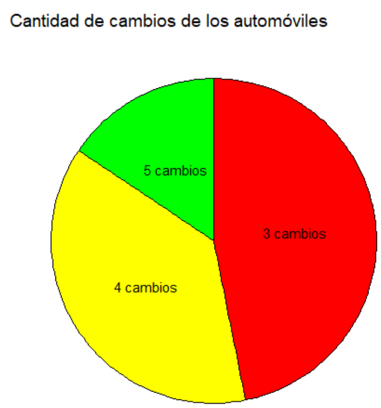


Figura 2.5: gráfico de torta para la variable `Cambios`.

Script 2.11: gráfico de torta para la variable `Cambios`.

```
library(ggpubr)

# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

# Crear la tabla de frecuencias y convertirla a data frame.
contingencia <- as.data.frame(xtabs(~ Cambios, data = datos))

# Crear gráfico de torta.
g <- ggpie(contingencia,
           x = "Freq",
           label = "Cambios",
           fill = c("red", "yellow", "green"),
           title = "Cantidad de cambios de los automóviles",
           lab.pos = "in")

print(g)
```

2.2.3 Dos variables numéricas

Los **gráficos de dispersión** son adecuados en este caso. Se caracterizan porque muestran información caso a caso, ya que cada punto del gráfico corresponde a una observación. Por ejemplo, el gráfico de la figura 2.6, creado mediante el script 2.12, muestra este tipo de gráfico para las variables `Rendimiento` y `Peso`.

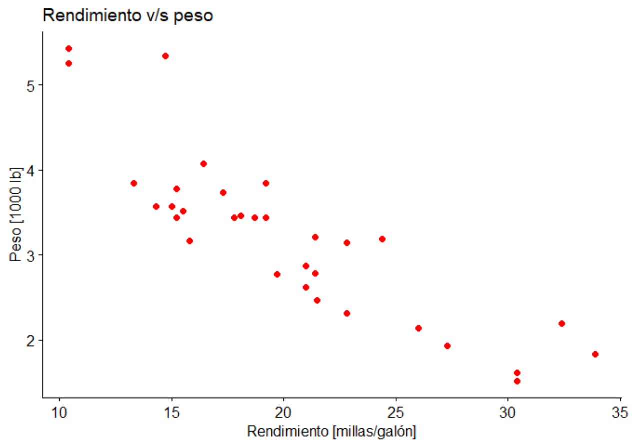


Figura 2.6: gráfico de dispersión para las variables `Rendimiento` y `Peso`.

Script 2.12: gráfico de dispersión para las variables `Rendimiento` y `Peso`.

```
library(ggpubr)

# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

# Crear gráfico de dispersión.
g <- ggscatter(datos,
               x = "Rendimiento",
               y = "Peso",
               color = "red",
               title = "Rendimiento v/s peso",
               xlab = "Rendimiento [millas/galón]",
               ylab = "Peso [1000 lb]")

print(g)
```

Los gráficos de dispersión también son muy útiles para identificar si dos (o más) variables están relacionadas. La figura 2.7 (creada mediante el script 2.13) muestra tres gráficos de dispersión diferentes: en el de la izquierda, se aprecia que las variables `Peso` y `Cuarto_milla` son independientes, pues no hay una tendencia definida en la organización de los puntos. En el gráfico del centro, en cambio, podemos ver que la potencia tiende a aumentar a medida que también lo hace el peso, por lo que ambas variables están positivamente asociadas. Por último, el gráfico de la derecha nos muestra que las variables `Peso` y `Rendimiento` presentan asociación negativa, puesto que a medida que la primera aumenta, la segunda disminuye.

Script 2.13: gráficos de dispersión con diferentes tipos de asociación entre las variables.

```
library(ggpubr)

# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)
```

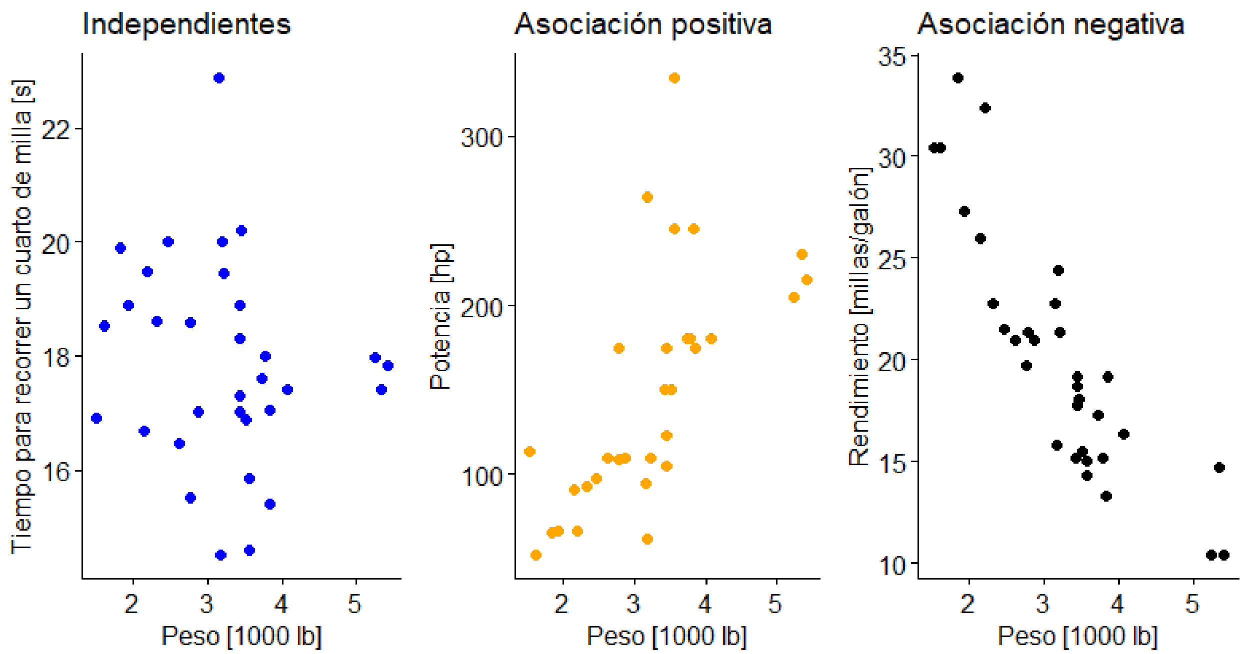


Figura 2.7: gráficos de dispersión con diferentes tipos de asociación entre las variables.

```r

# Gráfico para variables independientes.
g1 <- ggscatter(datos,
                x = "Peso",
                y = "Cuarto_milla",
                color = "blue",
                title = "Independientes",
                xlab = "Peso [1000 lb]",
                ylab = "Tiempo para recorrer un cuarto de milla [s]")

# Gráfico para variables con asociación positiva.
g2 <- ggscatter(datos,
                x = "Peso",
                y = "Potencia",
                color = "orange",
                title = "Asociación positiva",
                xlab = "Peso [1000 lb]",
                ylab = "Potencia [hp]")

# Gráfico para variables con asociación negativa.
g3 <- ggscatter(datos,
                x = "Peso",
                y = "Rendimiento",
                color = "black",
                title = "Asociación negativa",
                xlab = "Peso [1000 lb]",
                ylab = "Rendimiento [millas/galón]")

# Crear figura con tres gráficos.
g  <- ggarrange(g1 ,g2 ,g3, ncol = 3, nrow = 1, common.legend = TRUE)

print(g)
```

2.2.4 Dos variables categóricas

Similares al gráfico de barras para una variable categórica, los **gráficos de barras apiladas, agrupadas y estandarizadas** permiten visualizar la matriz de confusión entre dos variables y encontrar posibles relaciones entre ellas. La figura 2.8, creada con el script 2.14, ejemplifica esta familia de gráficos usando para ello las variables `Cambios` y `Motor`.

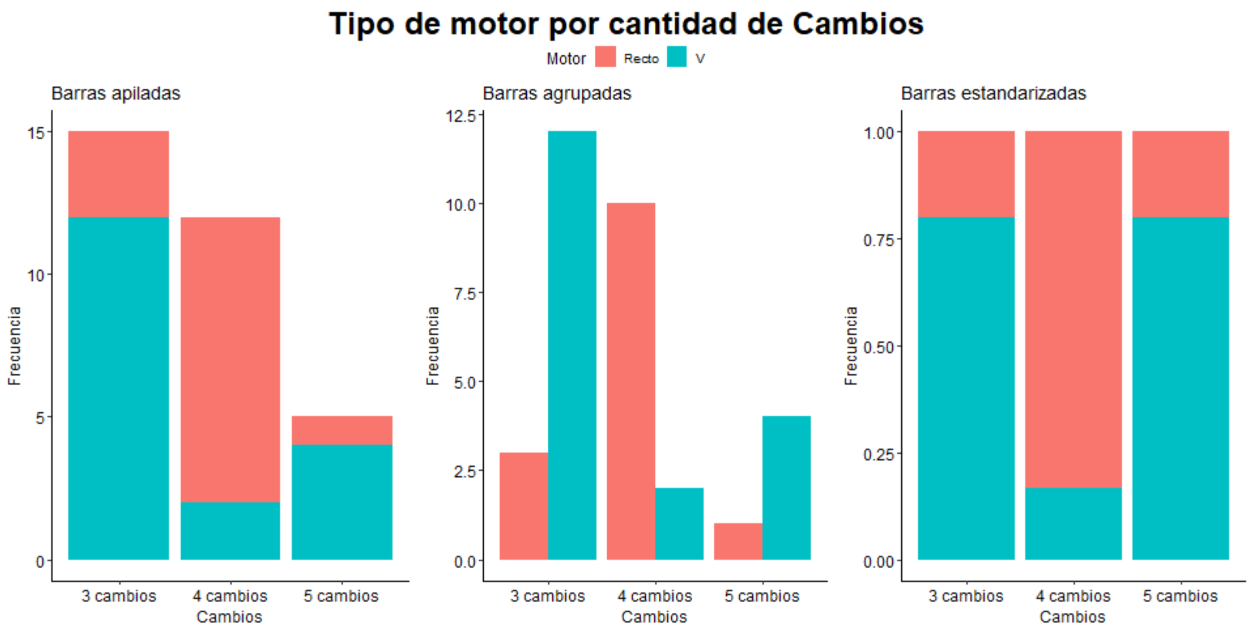


Figura 2.8: gráficos de barras para las variables `Cambios` y `Motor`.

El **gráfico de barras apiladas**, a la izquierda en la figura 2.8, muestra tres barras cuya altura corresponde a la frecuencia de la cantidad de cambios, al igual que en la figura 2.4, pero ahora cada barra está subdividida en secciones de distinto color para cada tipo de motor. La altura de cada sección está dada por la frecuencia del tipo de motor para la cantidad de cambios representada en la barra.

Similar al anterior, el gráfico de la derecha en la figura 2.8, que corresponde a un **gráfico de barras estandarizadas**, muestra barras de igual altura para cada cantidad de cambios, usando para ello la tabla de proporciones por columnas, representando claramente los cambios en la proporción de cada tipo de motor por la cantidad de cambios. Se puede apreciar que los automóviles con 3 y 5 cambios tienen mayoritariamente motores en forma de V, ambas en igual proporción, mientras que el uso de motores rectos se da principalmente en automóviles de 4 cambios.

El **gráfico de barras agrupadas**, al centro en la figura 2.8, es equivalente al de la izquierda, pero en lugar de dividir una barra en segmentos, muestra barras contiguas para cada tipo de motor.

Script 2.14: gráficos de barras para las variables `Cambios` y `Motor`.

```
library(ggpubr)

# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

# Crear tabla de contingencia para las variables Motor y Cambios,
# y guardarla como data frame.
tabla <- xtabs(~ Motor + Cambios, data = datos)
contingencia <- as.data.frame(tabla)
```

```

# Crear tabla de proporciones por columnas y guardarla como
# data frame.
proporciones <- as.data.frame(prop.table(tabla , margin = 2))

# Crear gráfico de barras segmentadas.
g1 <- ggplot(contingencia, aes(fill = Motor, y = Freq, x = Cambios))
g1 <- g1 + geom_bar(position = "stack", stat = "identity")
g1 <- g1 + labs(y = "Frecuencia") + ggtitle("Barras apiladas")
g1 <- g1 + theme_pubr()

# Crear gráfico de barras agrupadas.
g2 <- ggplot(contingencia, aes(fill = Motor, y = Freq, x = Cambios))
g2 <- g2 + geom_bar(position = "dodge", stat = "identity")
g2 <- g2 + labs(y = "Frecuencia") + ggtitle("Barras agrupadas")
g2 <- g2 + theme_pubr()

# # Crear gráfico de barras segmentadas estandarizado.
g3 <- ggplot(contingencia, aes(fill = Motor, y = Freq, x = Cambios))
g3 <- g3 + geom_bar(position = "fill", stat = "identity")
g3 <- g3 + labs(y = "Frecuencia") + ggtitle("Barras estandarizadas")
g3 <- g3 + theme_pubr()

# Crear una figura que contenga los tres gráficos.
g <- ggarrange(g1, g2, g3, nrow = 1, common.legend = TRUE)

# Agregar un título común en negrita y con fuente de 24 puntos.
titulo <- text_grob("Tipo de motor por cantidad de Cambios",
                    face = "bold", size = 24)

g <- annotate_figure(g, top = titulo)

# Guardar la figura en formato png con tamaño 960 x 480 pixeles.
ggexport(g, filename = "C:/Inferencia/f-barras-2.png", height = 480, width =
    960)
```

Similar al gráfico de barras para dos variables, el **gráfico de mosaico** permite representar una tabla de contingencia. Para ello, divide un área en regiones y el área de cada región es proporcional al porcentaje de observaciones que representa. La figura 2.9, creada con el script 2.15 ejemplifica el uso de este tipo de gráficos, usando para ello las variables `Cambios` y `Motor`. En ella, el ancho de cada columna es proporcional a la cantidad de automóviles que tienen la correspondiente cantidad de cambios, mientras que la altura de cada barra de las columnas refleja la proporción de automóviles con un determinado tipo de motor.

Si nos fijamos bien en la figura 2.9, podemos ver claramente que los vehículos con 5 cambios son, por mucho, los menos frecuentes y que los con 3 cambios son algo más frecuentes que los que tienen 4 cambios. Del mismo modo, podemos ver que, para vehículos de 3 y 5 cambios, la proporción de vehículos con motor recto es la misma, y mucho menor que la de aquellos con motor en forma de V. Sin embargo, este último no es muy frecuente en automóviles con 4 cambios.

Cabe destacar que, para este tipo de gráfico, se requiere emplear el paquete `ggmosaic`.

Script 2.15: gráfico de mosaico para las variables `Cambios` y `Motor`.

```
library(ggmosaic)

# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

# Crear tabla de contingencia para las variables gear y vs,
```

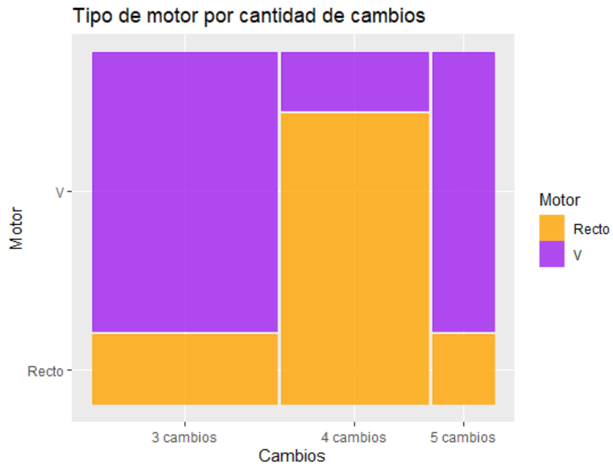


Figura 2.9: gráfico de mosaico para las variables `Cambios` y `Motor`.

```r
# y guardarla como data frame.
tabla <- xtabs(~ Cambios + Motor, data = datos)
contingencia <- as.data.frame(tabla)

# Crear gráfico de mosaico.
g <- ggplot(data = contingencia)
g <- g + geom_mosaic(aes(weight = Freq, x = product(Cambios), fill = Motor))

g <- g + labs(y = "Motor", x = "Cambios",
              title = "Tipo de motor por cantidad de cambios")

g <- g + scale_fill_manual(values=c("orange", "purple"))

print(g)
```

2.2.5 Una variable numérica y otra categórica

Desde luego, también es importante poder comparar diferentes grupos de observaciones de acuerdo a una característica categórica, para lo cual los gráficos pueden ser de gran ayuda. Por ejemplo, la figura 2.10, creada mediante el script 2.16 muestra un gráfico de cajas para la variable `Rendimiento` agrupada por el número de cambios de los automóviles.

Script 2.16: gráfico de cajas por grupo.

```r
library(ggpubr)

# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

g <- ggboxplot(datos, x = "Cambios",
          y = "Rendimiento",
          palette = c("light blue", "pink", "yellow"),
          fill = "Cambios",
```

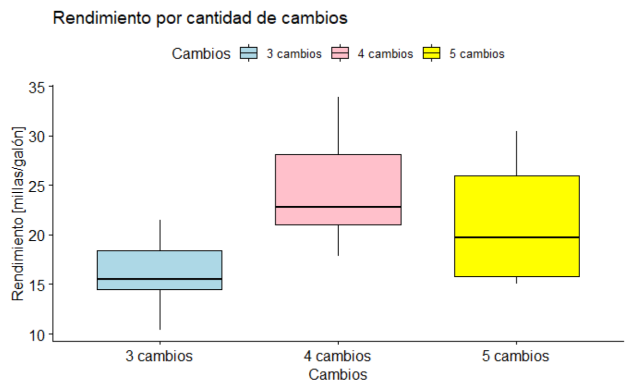


Figura 2.10: gráfico de cajas por grupo.

```
            title = "Rendimiento por cantidad de cambios",
            xlab = "Cambios",
            ylab = "Rendimiento [millas/galón]")

print(g)
```

Una buena alternativa, si la cantidad de observaciones es pequeña, es el **gráfico de tiras**, similar al gráfico de dispersión. El script 2.17 construye este gráfico para la variable `Rendimiento` agrupada según los niveles de la variable `Cambios`, obteniéndose como resultado la figura 2.11.

Script 2.17: gráfico de tiras.

```
library(ggpubr)

# Cargar datos.
datos <- read.csv2("C:/Inferencia/Mtcars.csv", stringsAsFactors = TRUE,
                   row.names = 1)

g <- ggstripchart(datos, x = "Cambios",
                  y = "Rendimiento",
                  palette = c("blue", "red", "dark green"),
                  color = "Cambios",
                  title = "Rendimiento por cantidad de cambios",
                  xlab = "Cambios",
                  ylab = "Rendimiento [millas/galón]")

print(g)
```

2.3 EJERCICIOS PROPUESTOS

1. ¿Cuándo es apropiado utilizar un gráfico de puntos para revisar datos?
2. ¿Cuándo la mediana caracteriza mejor a un conjunto de datos que la media?
3. Da ejemplos de tres variables que posiblemente tengan una distribución simétrica, con asimetría positiva y con asimetría negativa, respectivamente. Justifique bien cada caso.
4. Describe un estudio en que posiblemente los datos recolectados tengan una distribución bimodal.
5. ¿Qué tipo de información buscada llevaría a utilizar un gráfico de dispersión?

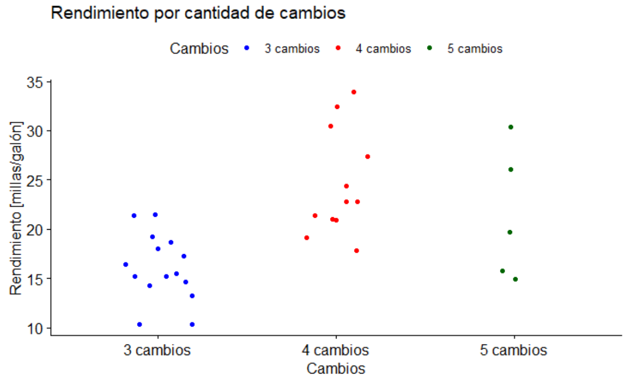
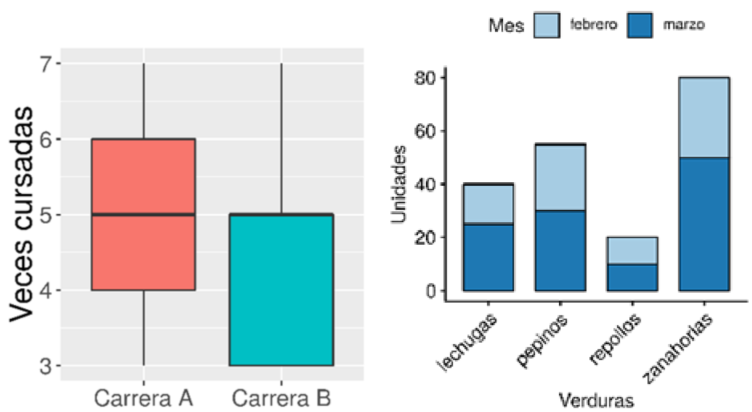


Figura 2.11: gráfico de tiras.

6. ¿Por qué es importante conocer una medida de dispersión (variabilidad) de un conjunto de datos? Dé un ejemplo para clarificar su respuesta.
7. Considera la representación de la figura 2.12a de los datos obtenidos al tomar muestras aleatorias de estudiantes de dos carreras de la Facultad de Ingeniería para estudiar si el número de veces que se cursan las tres asignaturas de física en el Módulo Básico de Ingeniería depende de la carrera de los alumnos. Compara las distribuciones de ambos grupos. ¿En qué se parecen y en qué se diferencian?



(a) Ejercicio 7.

(b) Ejercicio 8.

Figura 2.12: gráficos para los ejercicios propuestos.

8. El gráfico de la figura 2.12b muestra las unidades de verduras vendidas en uno de los kioskos de la Universidad durante los meses anteriores. Construye la tabla de contingencia correspondiente a los datos que se representan. ¿Qué mes tuvo mayores ventas? ¿En qué proporción?
9. ¿Cómo se puede generar la secuencia 0.00, 0.25, 0.50, ..., 2.75, 3.00 en R?
10. Resuelve en R los siguientes ejercicios. Considera para ello el conjunto de datos nativo de R `chickwts`.
    a) ¿Qué son los cuartiles y cómo se pueden obtener para los pesos de los pollitos reportados en la columna `weight`?
    b) ¿Cómo obtener los cuartiles del ejercicio anterior por cada tipo de alimento en la columna `feed`?
    c) ¿Cómo obtener un histograma de los pesos de los pollitos?
    d) ¿Cómo se obtiene un gráfico de cajas para comparar los pesos de los pollitos por tipo de alimento suministrado?
11. Investiga acerca del uso del paquete de R `ggplot2` para la creación de gráficos.

REFERENCIAS

Diez, D., Barr, C. D. & Çetinkaya-Rundel, M. (2017). *OpenIntro Statistics* (3.ª ed.). https://www.openintro.org/book/os/.

Field, A., Miles, J. & Field, Z. (2012). *Discovering statistics using R*. SAGE Publications Ltd.

STDHA. (s.f.). Descriptive Statistics and Graphics - Easy Guides - Wiki - STHDA. Consultado el 31 de marzo de 2021, desde http://www.sthda.com/english/wiki/descriptive-statistics-and-graphics