Lectura 2

Estadística descriptiva

-> Nos permiten sintetizar y resumir datos

-> Pueden aplicarse tanto a una muestra como a una población

-> Cuando estas medidas se aplican a la muestra, corresponde a un **estimador muestral** 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

Importante!!

Concepto de distribución, donde se considera **distribución de frecuencia,** que representa cuántas veces aparece cada valor para una variable en un conjunto de datos.

Estadística descriptiva 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()`.

```
# 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))
```

Una medida de tendencia central alternativa 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 como **median()**

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.

Para conocer la variabilidad o la dispersión, que indica que tan semejantes (o diferentes) son las observaciones entre sí.

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.

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

Las funciones 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.

Percentiles

->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.

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()`.

```
# 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()`.

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

La mediana resulta ser una buena medida de tendencia central y el IQR, una buena medida de dispersión.

->Summary() calcula varias medidas de tendencia central en una sola llamada

->Nos entrega la media, la mediana, el primer y el tercer cuartil, el mínimo y el máximo.

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

```r
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")
```

Estadística descriptiva para datos categóricos (Tabla de contingencia)

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")

```

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`.

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

```
# 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.

```r

# 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")
```

Datos agrupados

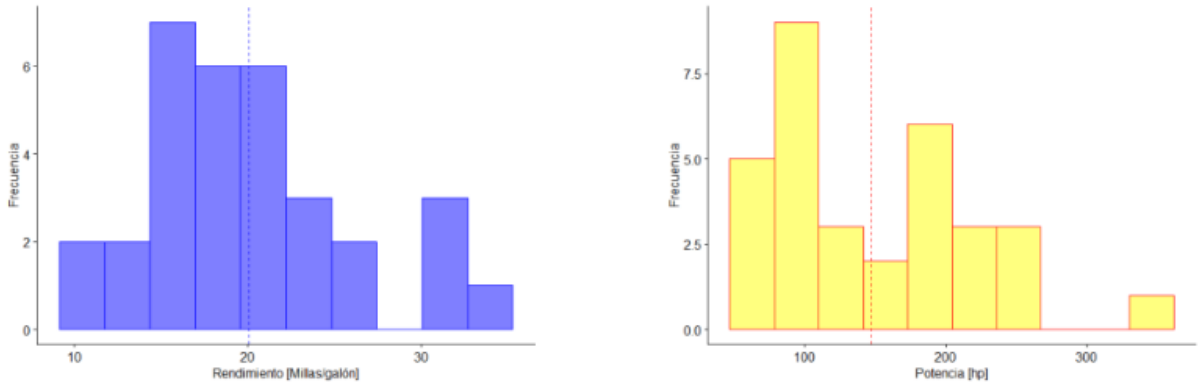
A menudo nos veremos en la necesidad de obtener estadísticas descriptivas de una variable separando las observaciones en grupos de acuerdo a una variable categórica. Para ello, el paquete `dplyr` ofrece la función `group_by()`, que podemos usar en conjunto con `summarise()`, como muestra el script 2.7. 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)
```

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)
```

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.

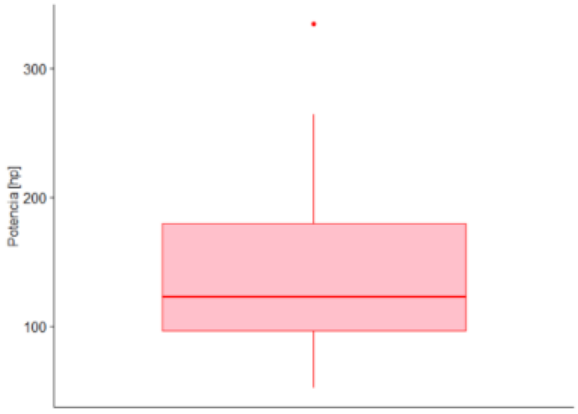


Figura 2.3: gráfico de caja para la variable `Potencia`.

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)
```

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.

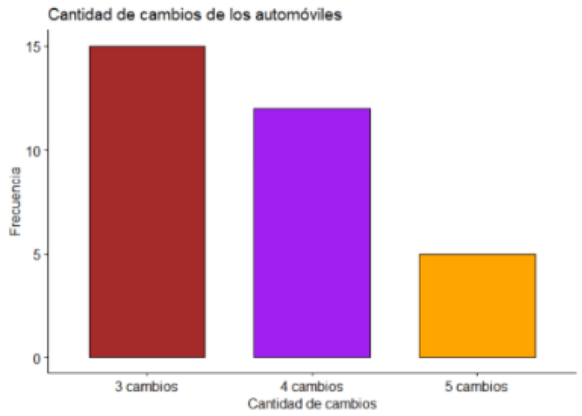


Figura 2.4: gráfico de barras para la variable `Cambios`.

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.

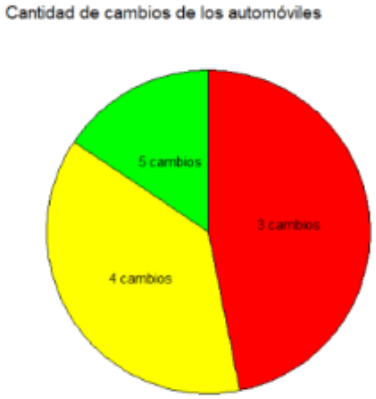


Figura 2.5: gráfico de torta para la variable `Cambios`.

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)
```

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`.

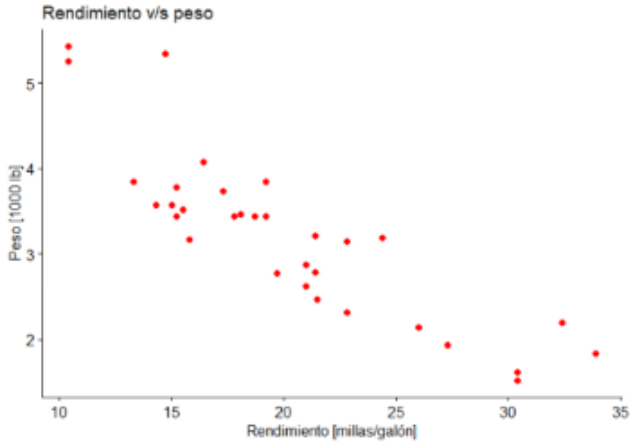


Figura 2.6: gráfico de dispersión para las variables `Rendimiento` y `Peso`.

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)
```

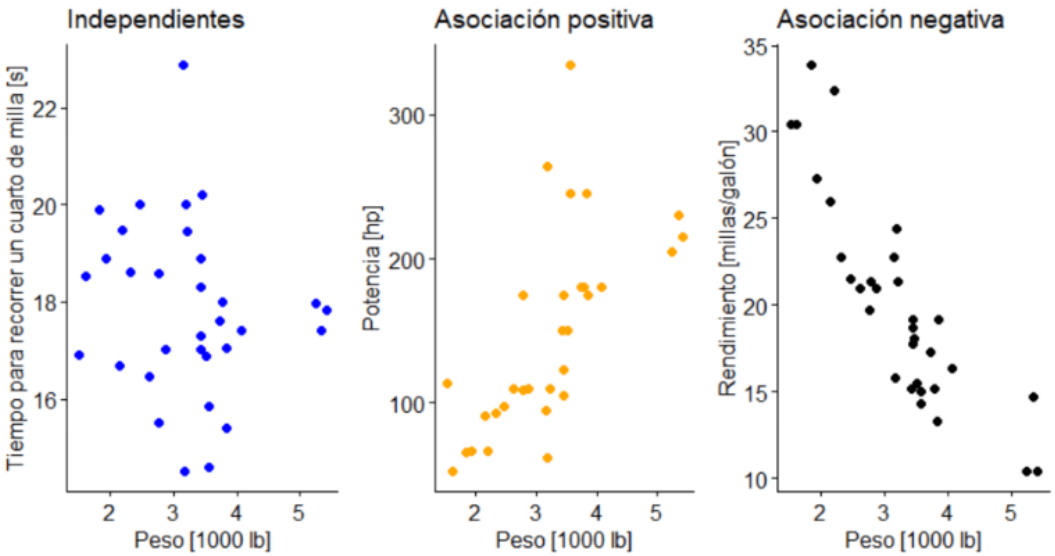


Figura 2.7: gráficos de dispersión con diferentes tipos de asociación entre las variables.

```
6
7  # Gráfico para variables independientes.
8  g1 <- ggscatter(datos,
9                  x = "Peso",
10                 y = "Cuarto_milla",
11                 color = "blue",
12                 title = "Independientes",
13                 xlab = "Peso [1000 lb]",
14                 ylab = "Tiempo para recorrer un cuarto de milla [s]")
15
16 # Gráfico para variables con asociación positiva.
17 g2 <- ggscatter(datos,
18                 x = "Peso",
19                 y = "Potencia",
20                 color = "orange",
21                 title = "Asociación positiva",
22                 xlab = "Peso [1000 lb]",
23                 ylab = "Potencia [hp]")
24
24
25 # Gráfico para variables con asociación negativa.
26 g3 <- ggscatter(datos,
27                 x = "Peso",
28                 y = "Rendimiento",
29                 color = "black",
30                 title = "Asociación negativa",
31                 xlab = "Peso [1000 lb]",
32                 ylab = "Rendimiento [millas/galón]")
33
34 # Crear figura con tres gráficos.
35 g  <- ggarrange(g1 ,g2 ,g3, ncol = 3, nrow = 1, common.legend = TRUE)
36
37 print(g)
```

Dos variables categóricas

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)
```

Una variable numérica y una 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.

```
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",
```

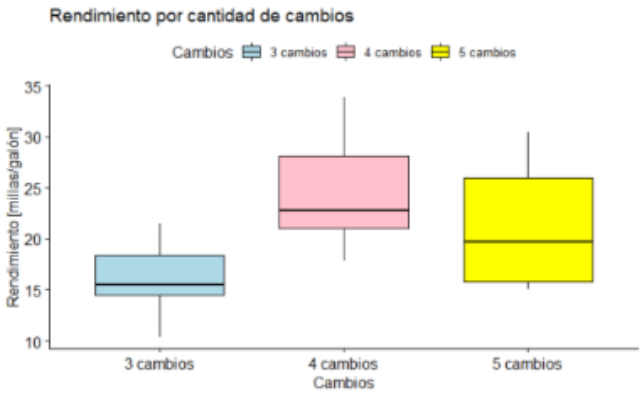


Figura 2.10: gráfico de cajas por grupo.

```
            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)
```