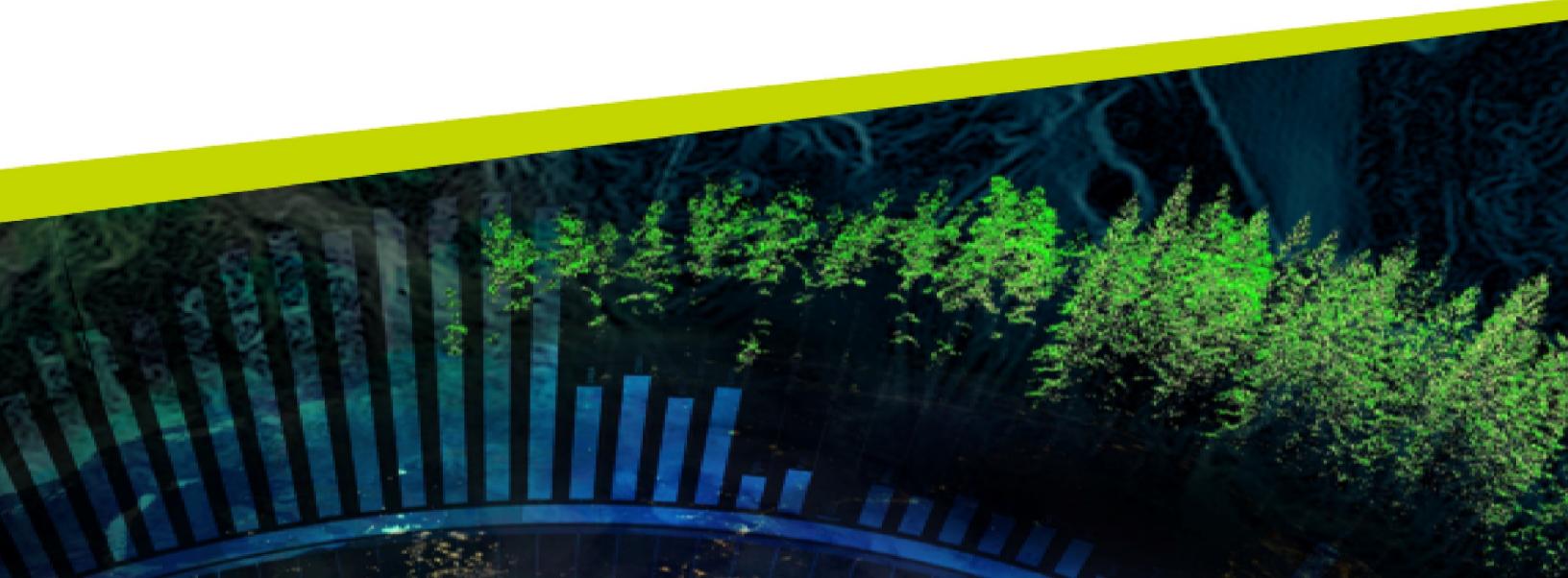




# INFERENCIA Y MODELOS ESTADÍSTICOS

Jacqueline Köhler C. y José Luis Jara V.



## CAPÍTULO 13. REGRESIÓN LINEAL

En el capítulo 2 introdujimos los gráficos de dispersión como una herramienta que nos permite identificar posibles relaciones entre dos variables cuantitativas. En este capítulo estudiaremos la **regresión lineal simple** (RLS), herramienta que sistematiza esta idea, basándonos en los textos de Diez et al. (2017, pp. 331-355), Field et al. (2012, pp. 245-311), Irizarry (2019, pp. 535-545) y Fox y Weisberg (2018, cap. 6).

La RLS asume que la relación entre dos variables aleatorias,  $X$  e  $Y$ , puede ser modelada mediante **una recta** de la forma que se presenta en la ecuación 13.1, donde:

- $\beta_0$  y  $\beta_1$  son los parámetros del modelo lineal.
- $x$  es una observación de la variable explicativa o **predictor** (variable independiente).
- $\hat{y}$  es una estimación del valor correspondiente de la variable **de respuesta** o **de salida** (variable dependiente).

$$\hat{y} = \beta_0 + \beta_1 x \quad (13.1)$$

Llamamos **intercepción** (*intercept*, en inglés) al parámetro  $\beta_0$ , que corresponde al punto en que la recta corta el eje  $y$ . A su vez, denominamos **pendiente** al parámetro  $\beta_1$ , el cual determina la inclinación de la recta del modelo.

Si tuviéramos una relación lineal **perfecta** entre ambas variables, significaría que se podríamos conocer el valor exacto de  $Y$  con solo conocer el valor de  $x$ . Sin embargo, como podemos apreciar en la figura 13.1, rara vez los datos se ajustan al modelo con exactitud.

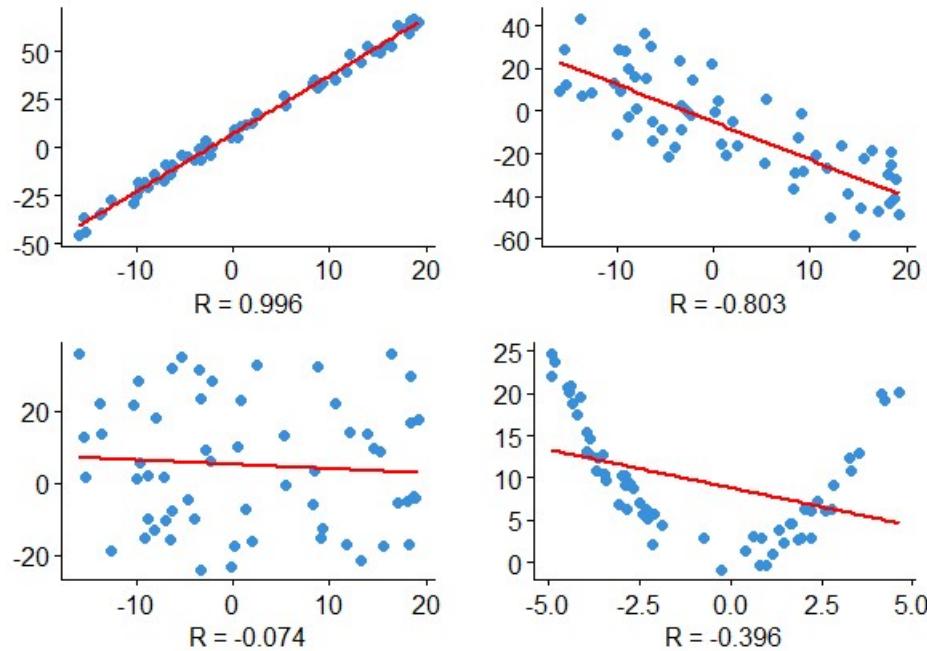


Figura 13.1: modelos lineales para cuatro conjuntos de datos.

Los gráficos de la fila superior de la figura 13.1 muestran dos tendencias lineales, siendo la izquierda una relación directa y muy fuerte, y la de la derecha, inversa y algo más débil. En el caso de los gráficos de la fila inferior, los datos en el de la izquierda no se aglutan en torno a la recta marcada, y los de la derecha, presentan un escenario donde ambas variables se relacionan clara y fuertemente, pero de manera no lineal. Siempre tenemos que tener en cuenta que, si los datos presentan una tendencia no lineal, debemos usar herramientas más avanzadas que la regresión lineal simple.

Fijémonos en el siguiente modelo lineal, que corresponde a la línea roja en el gráfico de arriba a la izquierda de la figura 13.1:

$$\hat{y} = 7 + 3x$$

En él, si  $x = 5$ , entonces  $\hat{y} = 22$ .  $\hat{y}$  es un estimador que podemos entender de la siguiente manera: dado un valor de  $x$ , el valor de  $y$  es, en promedio,  $\hat{y}$ . En otras palabras,  $\hat{y}$  corresponde al valor esperado de  $y$  para un determinado valor de  $x$ . En la práctica, existe una diferencia entre el valor esperado  $\hat{y}$  y el valor observado de  $y$ . Esta diferencia se denomina **residuo** y se denota  $e$  (del inglés *error*). Así, tenemos que el valor observado de  $y$  está dado por la ecuación 13.2.

$$y = \hat{y} + e \quad (13.2)$$

Otra forma de entender el residuo es como la distancia vertical que separa a la observación de la recta. Si la observación se encuentra por sobre esta última, entonces  $e > 0$ . En caso contrario,  $e < 0$ . Puesto que los residuos sirven para evaluar qué tan bien se ajusta un modelo lineal al conjunto de datos, suelen mostrarse en un **gráfico de residuos**, el cual es sencillamente un gráfico de dispersión donde la variable predictora se representa en su escala original y el eje  $y$  muestra el residuo para cada observación. La figura 13.2 muestra los residuos para los modelos lineales de la figura 13.1.

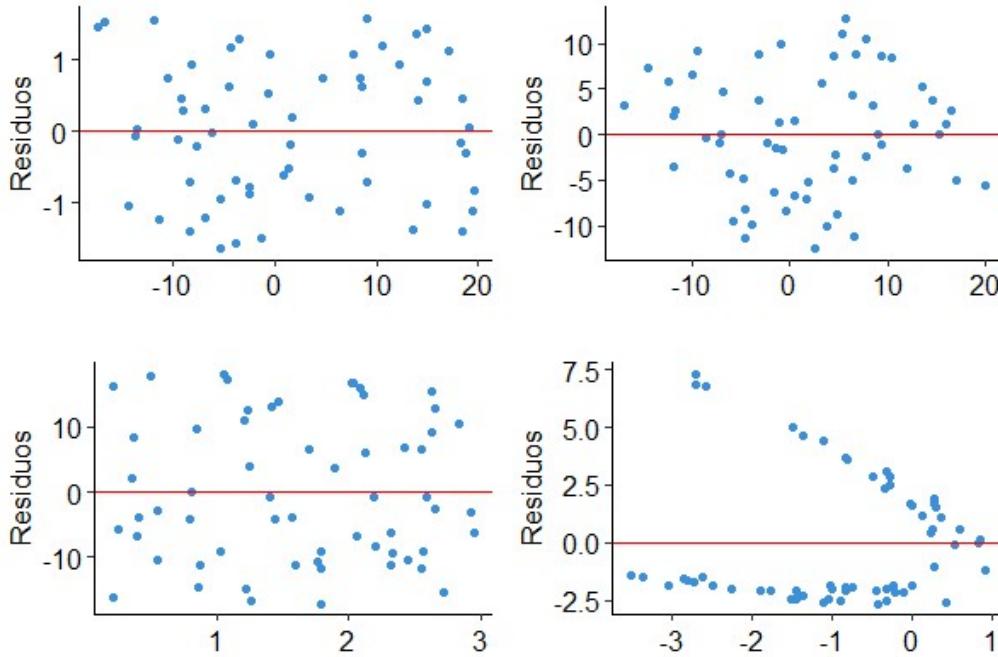


Figura 13.2: residuos para los modelos lineales de la figura 13.1.

## 13.1 CORRELACIÓN

Hasta ahora hemos hablado de la fuerza de una relación lineal entre dos variables, concepto que hemos asociado implícitamente a la magnitud de los residuos. Formalmente, podemos medir la fuerza de una relación lineal entre las variables  $X$  e  $Y$  mediante la **correlación**.

Una de las definiciones de correlación más usada, que incluso se usa como sinónimo cuando no se especifica explícitamente otra cosa, es el **coeficiente de correlación de Pearson**, que se define como la **covarianza** de las variables normalizada por la multiplicación de sus desviaciones estándar:

$$R = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{\sigma_{XY}}{\sqrt{\sigma_{XX}} \sqrt{\sigma_{YY}}} \quad (13.3)$$

Notemos que en esta ecuación hemos implícitamente usado la propiedad de la covarianza de una variable consigo misma (o dos variables iguales) corresponde a su varianza. Con esta normalización, la correlación siempre toma un valor entre -1 y 1. Mientras más débil sea la relación entre dos variables, su valor será más cercano a 0. El signo de la correlación indica si la relación es **directa** ( $R > 0$ ) o **inversa** ( $R < 0$ ). En el primer caso, cuando los valores de la variable  $X$  aumentan, se observan valores más altos para la variable  $Y$ ; mientras que en el segundo, cuando los valores de  $X$  van aumentando, los valores de la variable  $Y$  van disminuyendo.

Pensando en las muestras de dos variables, el coeficiente de correlación de Pearson queda dado por la ecuación 13.4, donde:

- $\bar{x}, \bar{y}$  son las medias de las variables  $X$  e  $Y$  en la muestra.
- $s_x, s_y$  corresponden a las desviaciones estándar de las variables  $X$  e  $Y$  en la muestra.
- $n$  es el tamaño de la muestra.

$$R = \frac{1}{n-1} \cdot \sum_{i=1}^n \frac{x_i - \bar{x}}{s_x} \cdot \frac{y_i - \bar{y}}{s_y} \quad (13.4)$$

Para comprender mejor idea, fíjemonos en los coeficientes de correlación obtenidos para cada modelo lineal de la figura 13.1, indicados en las etiquetas del eje  $x$ . Podemos ver que, entre más cerca estén las observaciones a la línea de RLS, más alto es el valor absoluto de  $R$  (gráficos superiores), mientras que dos variables que parecen no variar en conjunto llevan a valores de  $R$  cercanos a cero. En el caso del gráfico de abajo a la derecha, la relación es muy fuerte, pero no lineal, por lo que  $R$ , al considerar solo una recta, toma un valor relativamente bajo.

Para un ejemplo más concreto, que usaremos a lo largo de este capítulo, consideremos el conjunto de datos `mtcars`, disponible en R, que contiene diversas características para  $n = 32$  modelos de automóviles de los años 1973 y 1974. La tabla 13.1 describe brevemente cada una de las variables de dicho conjunto.

Columna	Descripción
mpg	Rendimiento, en millas (EEUU) por galón de gasolina [millas/galón].
cyl	Cantidad de cilindros del motor (4, 6 u 8).
disp	Volumen útil de los cilindros de un motor en pulgadas cúbicas [ $\text{in}^3$ ].
hp	Potencia del motor, en caballos de fuerza [hp].
drat	Relación de transmisión del eje trasero, en número de vueltas del eje de transmisión por cada rotación del eje de la rueda.
wt	Peso total, en miles de libras.
qsec	Tiempo mínimo para recorrer un cuarto de milla desde el reposo, en segundos [s].
vs	Tipo de motor (0: en forma de V, 1: recto).
am	Transmisión (0: automática, 1: manual).
gear	Número de marchas hacia adelante (3, 4 o 5).
carb	Número de carburadores (1, 2, 3, 4, 6 u 8).

Tabla 13.1: descripción de las variables para el conjunto de datos `mtcars` usados en este capítulo.

Consideremos la variable  $Y$ : potencia del motor (`hp`) en vehículos de entre 2 y 5 mil libras de peso (`wt >= 2 & wt <= 5`). Podemos calcular la correlación de esta variable con cualquiera de las otras variables numéricas del conjunto, suponiendo que presentan una relación lineal. Por ejemplo, podríamos considerar  $X_1$ : volumen útil de los cilindros del motor (`disp`), o  $X_2$ : rendimiento del vehículo (`mpg`). La figura 13.3 muestra gráficamente la relación de la variable de salida  $Y$  con las variables predictoras  $X_1$  (izquierda) y  $X_2$  (derecha). Considerando que:  $\bar{y} = 149,920$  y  $s_y = 66,825$ ,  $\bar{x}_1 = 227,608$ ,  $s_{x_1} = 97,988$ ,  $\bar{x}_2 = 227,608$  y  $s_{x_2} = 97,988$ , las correlación son:

$$R = \frac{1}{25-1} \cdot \sum_{i=1}^n \frac{x_i - 227,608}{97,988} \cdot \frac{y_i - 149,920}{66,825} = 0,744$$

$$R = \frac{1}{25-1} \cdot \sum_{i=1}^n \frac{x_i - 19,416}{4,294} \cdot \frac{y_i - 149,920}{66,825} = -0,752$$

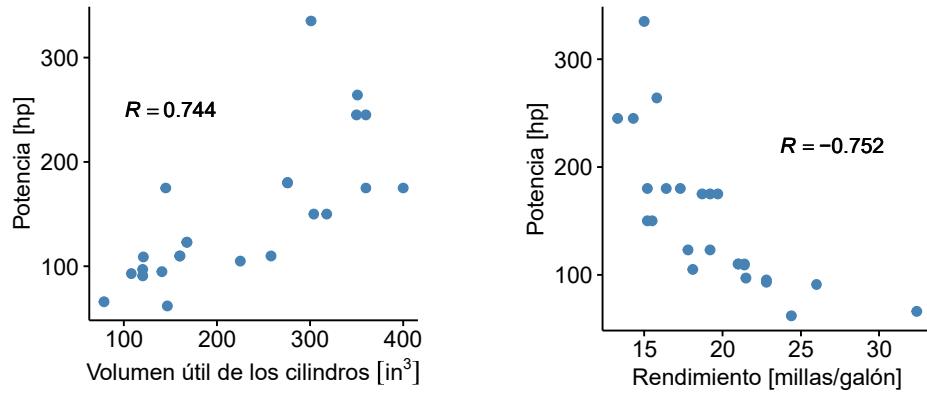


Figura 13.3: Relación entre el volumen útil de los cilindros (izquierda) y el rendimiento del vehículo (derecha) con la potencia del motor en vehículos que pesan entre 2 y 5 mil libras.

En R, podemos calcular la correlación entre dos variables usando la función `cor(x, y)`, donde `x` es el predictor e `y` la respuesta. Para el ejemplo, esta función devuelve que  $R = 0,74394$  y  $R = -0,75197$ , lo que coincide con los cálculos anteriores teniendo en cuenta que la diferencia se debe únicamente al redondeo.

Adicionalmente, cuando `x` es una matriz de datos, la función `cor(x)` nos entrega una **matriz de correlación**, que contiene las correlaciones entre todas los pares de columnas. La figura 13.4 muestra la matriz de correlación (redondeada al tercer decimal) que se obtiene para las variables numéricas del conjunto de datos `mtcars`. Podemos ver que, naturalmente, la matriz de correlación es simétrica y que su diagonal solo contiene unos.

	mpg	disp	hp	drat	wt
mpg	1.000	-0.762	-0.752	0.520	-0.782
disp	-0.762	1.000	0.744	-0.610	0.781
hp	0.744	1.000	-0.228	0.572	
drat	0.520	-0.610	-0.228	1.000	-0.675
wt	-0.782	0.781	0.572	-0.675	1.000

Figura 13.4: matriz de correlación para las variables numéricas del conjunto de datos `mtcars` en vehículos que pesan entre 2 y 5 mil libras

## 13.2 REGRESIÓN LINEAL MEDIANTE MÍNIMOS CUADRADOS

Si bien existen diversos métodos para ajustar un modelo lineal, el más empleado es el de la **línea de mínimos cuadrados**, que minimiza la suma de los cuadrados de los residuos (ecuación 13.5).

$$\min \sum_{i=1}^n e_i^2 \quad (13.5)$$

El método de mínimos cuadrados tiene las ventajas de ser fácil de calcular y de tomar en cuenta la discrepancia entre la magnitud del residuo y su efecto. Como señalan Diez et al. (2017, p. 341), “por ejemplo, desviarse por 4 suele ser más de dos veces peor que desviarse por 2”. No obstante, para aplicar este método debemos verificar que se cumplan algunas condiciones:

1. Las variables presentan una distribución condicional bivariante, por lo que, para cualquier valor fijo de  $X$ , los valores de  $Y$  se distribuyen normalmente con una varianza constante.
2. La relación entre la variable  $X$  y las medias de la variable  $Y$  es lineal.
3. Las observaciones de la muestra son independientes entre sí. Esto significa que no se puede usar regresión lineal con series de tiempo (tema que va más allá de los alcances de este texto).

Si bien estas condiciones son exigentes y muchas veces poco realistas con datos reales, el método de los mínimos cuadrados puede generar modelos completamente inválidos si estas no se cumplen. Más adelante veremos procedimientos de diagnóstico para evaluar su realización.

El primer paso que debemos seguir cuando queremos determinar la recta de mínimos cuadrados para un conjunto de datos consiste en estimar la pendiente ( $\beta_1$ ) mediante la ecuación 13.6, donde:

- $s_x$  y  $s_y$  son las desviaciones estándares muestrales de las variables  $X$  e  $Y$ , respectivamente.
- $R$  corresponde a la correlación entre ambas variables.

$$b_1 = \frac{s_y}{s_x} \cdot R \quad (13.6)$$

El punto  $(\bar{x}, \bar{y})$ , donde  $\bar{x}$  e  $\bar{y}$  son las medias muestrales para las variables  $X$  e  $Y$  respectivamente, siempre pertenece a la recta de mínimos cuadrados, por lo que podemos calcular la intercepción mediante la ecuación 13.7 (Winner, 2021, p. 4).

$$b_0 = \bar{y} - b_1 \cdot \bar{x} \quad (13.7)$$

Cuando contamos con más de una variable para construir una RLS, lo más adecuado es que escojamos como predictor aquella variable que muestre una clara relación lineal y una fuerte correlación con la variable de respuesta. Observando la figura 13.4 podemos ver que las variables `disp` y `mpg` son las que presentan los mayores valores de correlación con la variable `hp`. Por otro lado, los gráficos de la figura 13.3 muestran que existe una relación directa e inversa, respectivamente, de los predictores con la variable de salida, aunque claramente no son líneas perfectas.

Obtengamos los modelos lineales para estos predictores, comenzando con el cálculo de la pendiente de las rectas:

$$\begin{aligned} X_1 : \quad b_1 &= \frac{66,825}{97,988} \cdot 0,744 = 0,507 \\ X_2 : \quad b_1 &= \frac{66,825}{4,294} \cdot -0,752 = -11,703 \end{aligned}$$

A su vez, las intercepciones están dadas por:

$$\begin{aligned} X_1 : \quad b_0 &= 149,920 + 0,507 \cdot 227,608 = 34,523 \\ X_2 : \quad b_0 &= 149,920 - 11,703 \cdot 19,416 = 337,145 \end{aligned}$$

Por lo que la recta ajustada mediante mínimos cuadrados es:

$$\begin{aligned} X_1 : \quad \widehat{hp} &= 34,523 + 0,507 \text{ disp} \\ X_2 : \quad \widehat{hp} &= 337,145 - 11,703 \text{ mpg} \end{aligned}$$

Desde luego, R ofrece una función que permite ajustar la recta de mínimos cuadrados para un par de variables: `lm(formula, data)`, donde:

- `formula`: tiene la forma `<variable de respuesta> ~ <variable predictora>`.
- `data`: matriz de datos.

El script 13.1 ajusta la línea de mínimos cuadrados para la variable de respuesta potencia (`hp`), con el volumen útil de los cilindros (`disp`) como predictor, mediante el uso de `lm()` (línea 8). Un detalle interesante es que, en la línea 9 del script, usamos la llamada `print(summary(modelo))` en lugar de `print(modelo)`, lo que nos entrega información más detallada del modelo ajustado, el que es presentado en la figura 13.5. En esta última vemos que los coeficientes obtenidos para el modelo aparecen bajo el encabezado “`Coefficients`”,

donde podemos ver que los valores estimados para los parámetros de la RLS (bajo “Estimate”) coinciden con los calculados previamente.

Pero debemos notar que la función `lm()` nos entrega información que no teníamos sobre estos coeficientes. En primer lugar nos indica una estimación del error estándar para cada uno, bajo el título “Std. Error”. Con este valor es posible normalizar el coeficiente en términos de cuántas desviaciones estándar se aleja del valor cero, el que aparece bajo el título “t value”. Luego la función reporta la probabilidad de observar valores iguales o superiores a los coeficientes estandarizados en una distribución t con  $n - 2$  grados de libertad, siendo  $n$  el número de observaciones usadas en la construcción del modelo de RLS (con dos parámetros), bajo la etiqueta “Pr(>|t|)”.

Recordemos que, por lo que vimos en las figuras iniciales y al definir correlación, cuando no existe una relación entre dos variables, se espera encontrar un modelo LRS que corresponde a una línea recta horizontal. Es decir, esperamos que el coeficiente estimado  $b_1$  se aproxime a cero y que el intercepto se aproxime a la media muestral de la variable de salida ( $\hat{y} = \bar{y}$ ). De esta forma, los valores reportados bajo “t value” y “Pr(>|t|)” corresponden, respectivamente, al estadístico y valor p de la hipótesis nula que el coeficiente tiene valor cero:  $H_0 : \beta_i = 0$ .

Aplicando esta idea al ejemplo de la figura 13.5, podemos concluir que es posible rechazar esta hipótesis nula para el predictor `disp` ( $t(23) = 5,339$ ;  $p < 0,001$ ), por lo que parece tener una relación con la variable de salida `hp`.

```

Call:
lm(formula = hp ~ disp, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-62.382 -38.677   1.624   5.630 147.845 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 34.44423  23.47413   1.467   0.156    
disp         0.50734   0.09503   5.339 2.02e-05 *** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 45.62 on 23 degrees of freedom
Multiple R-squared:  0.5534, Adjusted R-squared:  0.534  
F-statistic: 28.5 on 1 and 23 DF,  p-value: 2.02e-05

```

Figura 13.5: detalles del un modelo de RLS obtenido para predecir la potencia de un automóvil a partir del volumen útil de los cilindros.

El gráfico de la izquierda de la figura 13.6 muestra la recta ajustada para el ejemplo construido en las líneas 12–18 del script 13.1. Podemos apreciar que los datos presentan una relación lineal, aunque algunos puntos parecen estar algo alejados de la recta ajustada.

Script 13.1: ajuste de una regresión lineal simple.

```

1 library(dplyr)
2 library(ggpubr)
3
4 # Cargar y filtrar los datos.
5 datos <- mtcars |> filter(wt > 2 & wt < 5)
6
7 # Ajustar modelo con R.
8 modelo <- lm(hp ~ disp, data = datos)

```

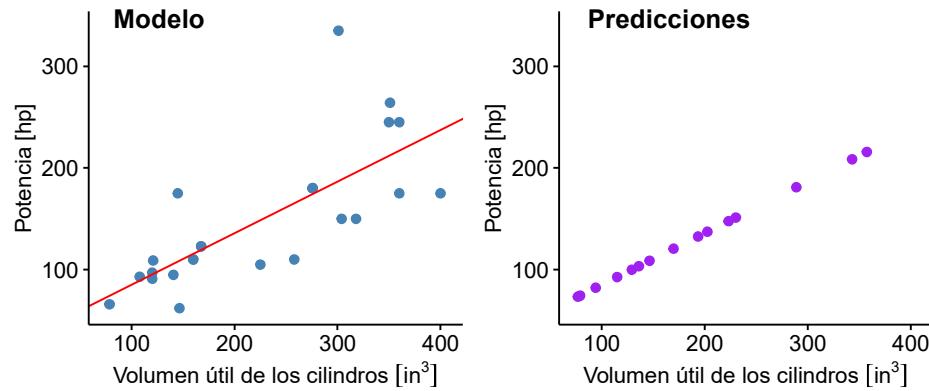


Figura 13.6: modelo de RLS (izquierda) para la potencia de un automóvil y el volumen útil de los cilindros del motor en vehículos que pesan entre 2 y 5 mil libras, y las predicciones que entrega (derecha) para un conjunto de nuevos valores.

```

9 print(summary(modelo))
10
11 # Graficar los datos y el modelo obtenido
12 g1 <- ggscatter(datos, x = "disp", y = "hp",
13                  color = "steelblue", fill = "steelblue",
14                  ylab = "Potencia [hp]")
15 g1 <- g1 + geom_abline(intercept = coef(modelo)[1],
16                         slope = coef(modelo)[2],
17                         color = "red")
18 g1 <- g1 + xlab(bquote("Volumen útil de los cilindros" ~ group("[", "in"^3, "]")))
19
20 # Definir valores del predictor para vehículos no incluidos
21 # en el conjunto mtcars
22 disp <- c(169.694, 230.214, 79.005, 94.085, 343.085,
23          136.073, 357.305, 288.842, 223.128, 129.217,
24          146.432, 193.474, 376.874, 202.566, 114.928)
25
26 # Usar el modelo para predecir el rendimiento de estos modelos.
27 potencia_est <- predict(modelo, data.frame(disp))
28
29 # Graficar los valores predichos
30 nuevos <- data.frame(disp, hp = potencia_est)
31 g2 <- ggscatter(nuevos, x = "disp", y = "hp",
32                  color = "purple", fill = "purple",
33                  ylab = "Potencia [hp]")
34 g2 <- g2 + xlab(bquote("Volumen útil de los cilindros" ~ group("[", "in"^3, "]")))
35
36 # Unir los gráficos en uno solo
37 g1 <- ggpar(g1, xlim = c(75, 405), ylim = c(60, 340))
38 g2 <- ggpar(g2, xlim = c(75, 405), ylim = c(60, 340))
39 g <- ggarrange(g1, g2,
40                  labels = c("Modelo", "Predicciones"),
41                  hjust = c(-1.2, -0.7))
42 print(g)

```

### 13.3 USO DEL MODELO

Una de las etapas más importantes en un proceso de análisis es la **interpretación de los parámetros** del modelo. La pendiente explica la **diferencia esperada** en el valor de la variable respuesta  $Y$  si el predictor  $X$  se incrementa en una unidad. Así, para el ejemplo, se espera que al incrementar en una pulgada cúbica el volumen útil de los cilindros en vehículos que pesan entre 2 y 5 mil libras, la potencia del motor aumente en promedio 0,50 caballos de fuerza.

A su vez, la intercepción corresponde a la respuesta que se obtendría en promedio si  $x$  fuese igual a 0, suponiendo que el modelo fuese válido para  $x = 0$ , lo que no siempre ocurre. De hecho, para nuestro ejemplo, es imposible que un automóvil (a combustible, fabricado en los años 70) carezca de volumen en sus cilindros.

El párrafo anterior ilustra una limitación propia de cualquier modelo: este, al ser una simplificación de la realidad, tiene validez únicamente en el rango de valores de los datos originales, por lo que la **extrapolación** (es decir, estimar valores fuera del rango de los datos originales) puede conllevar a errores al asumir que el modelo es válido donde aún no ha sido analizado. El modelo del ejemplo, el predictor varía entre 78 y 400 [ $\text{in}^3$ ] aproximadamente, por lo que si lo usáramos para predecir la potencia de un vehículo cuyo volumen útil de los cilindros fuera de solo 39 [ $\text{in}^3$ ] (como el Fiat 600 que se fabricaba en Chile por esos años), el resultado podría carecer de validez.

Desde luego, debemos tener en cuenta también las condiciones de diseño. El resultado podría ser equivocado si con este modelo intentáramos predecir, por ejemplo, la potencia de un motor moderno, aun cuando la variable predictora tuviera valores dentro del rango utilizado (recordemos que el conjunto de datos contiene vehículos de los años 1973 y 1974). Más aún, la variable de respuesta carece absolutamente de sentido si pensamos, por ejemplo, en automóviles eléctricos (que no tienen cilindros).

Supongamos que queremos predecir la potencia de un motor del año 1974 con una cilindrada de 4 litros (es decir, `disp = 244.095`). Para ello, basta con reemplazar el valor del predictor en el modelo:

$$\widehat{\text{hp}} = 34,523 + 0,507 \cdot 244,095 = 158,279$$

Con lo que podemos predecir que el motor de tal vehículo tiene una potencia promedio de 158,3 caballos de fuerza.

En R, la función `predict(object, newdata)` nos permite usar un modelo (en este caso, una RLS obtenida con la función `lm()`) para predecir respuestas. Los argumentos de esta función son:

- `object`: el modelo a emplear.
- `newdata`: una matriz de datos donde exista una columna con el nombre del predictor usado en la fórmula del modelo (para el ejemplo, `disp`) con los nuevos valores para los que se desea efectuar la predicción.

La línea 27 del script 13.1 ilustra el uso de esta función para un conjunto de 15 valores generados artificialmente (líneas 22–24). Las predicciones obtenidas se muestran en el gráfico de la derecha de la figura 13.6, que es generado en las líneas 30–34 del script<sup>1</sup>. Podemos notar que los valores predichos siguen perfectamente la línea de regresión, pues esta representa el promedio de la variable de salida para cada valor del predictor.

### 13.4 REGRESIÓN LINEAL CON UN PREDICTOR CATEGÓRICO

Las variables categóricas también nos pueden servir para predecir una respuesta. En este capítulo solo estudiaremos el caso de una variable dicotómica (es decir, con solo dos niveles), idea que profundizaremos en el siguiente capítulo.

Para usar una variable categórica con dos niveles, tenemos que convertirla a formato numérico, para lo cual creamos una nueva **variable indicadora** que toma los valores 0 y 1. Hacer este proceso en R es bastante sencillo, y en la práctica, rara vez tendremos que realizar este paso manualmente, pues las funciones de R que ajustan modelos lo hacen automáticamente cuando encuentran predictores categóricos.

<sup>1</sup>El resto del script permite alinear los ejes de ambas figuras y combinarlas en una sola.

El conjunto de datos `mtcars` ya cuenta con un par de variables indicadoras: la transmisión (`am`) y la forma del motor (`vs`). De estas dos variables, la forma del motor tiene una correlación más fuerte con la potencia del motor, por lo que la usaremos como ejemplo para crear un modelo RLS. Al crear el modelo (script 13.2) obtenemos como resultado la recta representada en el gráfico izquierdo de la figura 13.7, con los residuos mostrados en el gráfico derecho de la misma figura. A su vez, la figura 13.8 muestra los valores obtenidos para los parámetros del modelo.

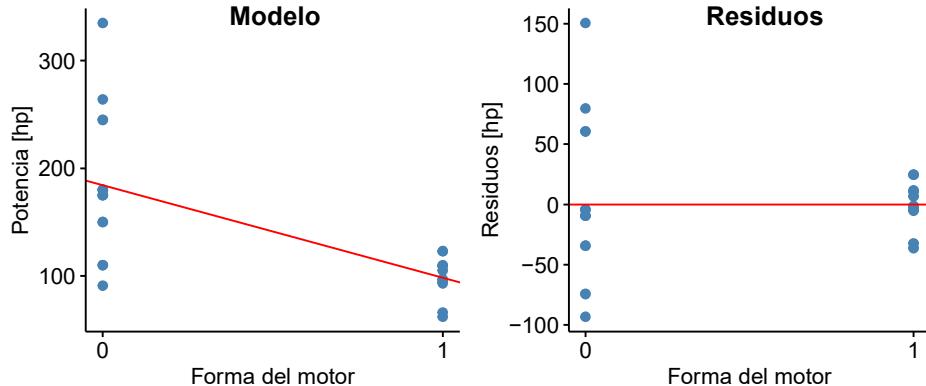


Figura 13.7: modelo de regresión lineal y gráfico de residuos para el ejemplo con un predictor dicotómico.

```

hp           am           vs
hp  1.000000000  0.004734644 -0.6437214
am  0.004734644  1.000000000 -0.1020621
vs  -0.643721448 -0.102062073  1.0000000

Call:
lm(formula = hp ~ vs, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-93.333 -32.300 -4.333  11.700 150.667 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 184.33     13.49   13.667 1.58e-12 ***
vs          -86.03     21.33   -4.034 0.000517 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 52.24 on 23 degrees of freedom
Multiple R-squared:  0.4144, Adjusted R-squared:  0.3889 
F-statistic: 16.27 on 1 and 23 DF,  p-value: 0.0005168

```

Figura 13.8: regresión lineal simple para el ejemplo con un predictor dicotómico.

Script 13.2: regresión lineal simple con un predictor dicotómico.

```

1 library(ggpubr)
2
3 # Obtener los datos.
4 datos <- mtcars |> filter ( wt > 2 & wt < 5)
5
6 # Verificar correlación.

```

```

7 print(cor(datos[, c("hp", "am", "vs")]))
8
9 # Ajustar modelo con R.
10 modelo_vs <- lm(hp ~ vs, data = datos)
11 print(summary(modelo_vs))
12
13 # Graficar el modelo.
14 g1 <- ggscatter(datos, x = "vs",
15                   color = "steelblue", fill = "steelblue",
16                   xlab = "Forma del motor", ylab = "Potencia [hp]",
17                   xticks.by = 1)
18 g1 <- g1 + geom_abline(intercept = coef(modelo_vs)[1],
19                         slope = coef(modelo_vs)[2],
20                         color = "red")
21 print(g1)
22
23 # Graficar residuos.
24 residuos <- modelo_vs[["residuals"]]
25 datos <- cbind(datos, residuos)
26
27 g2 <- ggscatter(datos, x = "vs", y = "residuos",
28                   color = "steelblue", fill = "steelblue",
29                   xlab = "Forma del motor", ylab = "Residuos [hp]",
30                   xticks.by = 1)
31 g2 <- g2 + geom_hline(yintercept = 0, color = "red")
32
33 # Unir los gráficos en uno solo
34 g <- ggarrange(g1, g2,
35                  labels = c("Modelo", "Residuos"),
36                  hjust = c(-2.5, -2.0))
37 print(g)

```

## 13.5 CONFIABILIDAD DE UN MODELO DE RLS

Hasta ahora hemos visto cómo ajustar y usar un modelo de RLS. Sin embargo, no hemos verificado que el modelo conseguido cumple con todas las condiciones para utilizar RLS. Tampoco hemos revisado si el modelo conseguido representa bien los datos observados. Es importante realizar estas verificaciones **antes** de utilizar un modelo de RLS, puesto que si se está violando alguna condición o no se ajusta a los datos, no es posible **confiar** en las predicciones que nos entrega.

### 13.5.1 Bondad de ajuste

Si observamos los modelos de RLS con un buen nivel de correlación (en las figuras 13.1 o 13.6, por ejemplo), podremos notar que conociendo el valor de la variable  $X$  podemos determinar una **mejor estimación** del valor de  $Y$  que simplemente usar la media muestral  $\bar{y}$ . Es decir, esperamos que la RLS nos entregue un **mejor modelo** de la variable  $Y$  que  $\bar{y}$ .

Es más, esta mejora se puede cuantificar a través de la **reducción en la varianza** que se obtiene con la recta de regresión en comparación a la recta horizontal. Si  $SS_{RLS}$  denota la suma de las desviaciones cuadradas de las estimaciones producidas por un modelo de RLS para una muestra de datos, y  $SS_{total}$  a la suma de las desviaciones cuadradas al usar la media muestral, entonces se puede definir el **coeficiente de determinación** como muestra la ecuación 13.8.

$$R^2 = \frac{SS_{total} - SS_{RLS}}{SS_{total}} \quad (13.8)$$

Podemos ver que el coeficiente de determinación corresponde a la proporción en que un modelo de RLS reduce la varianza global no explicada, por lo que se usa como una medida de la **bondad del ajuste** que consigue con los datos observados. Más aún, si tanto el numerador como el denominador de este coeficiente se dividen por los grados de libertad correspondientes, 1 y  $n - 2$  respectivamente, se puede obtener un estadístico  $F$ , y en consecuencia un valor  $p$ , para determinar si esta reducción de variabilidad es significativa.

Hagamos explícitas algunas ideas que podrían pasarse por alto. Primero, que la notación  $R^2$  no es accidental, ya que el coeficiente de determinación efectivamente corresponde al cuadrado de la correlación (Glen, 2021), por lo que “R-cuadrado” es un nombre alternativo bastante utilizado. Segundo, el valor de este coeficiente, al tratarse de una proporción, varía entre 0 y 1. Por último, notemos que  $SS_{RLS}$  no es otra cosa que la suma de los cuadrados de los residuos que el modelo produce para los datos observados.

Para el ejemplo, como podemos verificar en las últimas líneas de la información del modelo obtenido (figura 13.5), la función `lm()` reporta el coeficiente de determinación con el nombre de `Multiple R-squared`, igual a 0,553. Es decir, el predictor `disp` logra reducir la varianza aleatoria en un 55,3% respecto del modelo nulo (recta horizontal). En la última línea de la figura, vemos que esta reducción en la varianza es muy grande ( $F(1, 23) = 28,5$ ) y significativa ( $p < 0,001$ ).

### 13.5.2 Distribución e independencia

En general, las condiciones para aplicar RLS mencionadas en la sección 13.2 no son fáciles de verificar directamente. Afortunadamente, cuando estas condiciones se cumplen, se observan ciertas características en el gráfico de los residuos (Pardoe et al., 2018):

1. Se distribuyen aleatoriamente en torno a la línea de valor cero.
2. Forman una “banda horizontal” en torno a la línea de valor cero.
3. No hay residuos que se alejen del patrón que forman los demás.
4. No forman un patrón reconocible.

Las primeras tres características se dan cuando existe una distribución condicional normal bivariante entre el predictor y la variable de salida. Si se observan las características 1 y 4, es razonable suponer que las variables presentan una relación lineal y que las observaciones son independientes entre sí.

El paquete `car` ofrece varias funciones para realizar el diagnóstico de modelos de regresión. Entre ellas, se encuentra la función `residualPlots(modelo, type = "pearson")` que despliega gráficos de residuos, donde:

- `modelo`: el modelo a emplear.
- `type`: el tipo de residuo a utilizar. Por defecto toma valor `"Pearson"` que corresponde a la definición que hemos visto y graficado anteriormente. Sin embargo, estos residuos se encuentran en la escala de la variable de salida, por lo que a veces es difícil reconocer valores “muy alejados” de la línea del cero. Por esto existen otras opciones, como el tipo `"standard"`, que los presenta normalizados en términos de la desviación estándar presente en los datos, o el tipo `"studentized"`, que los presenta normalizados en términos de la desviación estándar sin considerar el residuo en cuestión. Sabemos que valores fuera del rango  $\pm 2$  (que representa aproximadamente el 95% de los datos en distribuciones normales o  $t$ ) podrían tratarse de valores atípicos.

Esta, y las otras funciones del paquete, tienen varios otros argumentos. Muchos de ellos se deben a que usa las funciones básicas para graficar (y no las basadas en el paquete `ggplot2` que utilizamos en este libro). Algunos de estos los iremos explicando a medida que se usan, pero el detalle de todos ellos debe encontrarse en la documentación oficial del paquete `car`. También es importante saber que estas funciones, que se ejecutan correctamente en la consola o en la línea principal de un script (en el ambiente global), pueden tener problemas al ser llamadas desde una función que nosotros construyamos, pues las reglas de alcance de R podrían impedir el acceso a los datos que ellas esperan.

El gráfico de diagnóstico más común para un modelo de RLS es el gráfico de los residuos en función de los valores predichos por el modelo para las observaciones, que suelen llamarse los valores “ajustados” (*fitted* en inglés). Sin embargo, se suele revisar también el gráfico de residuos en función de los valores del predictor. Si el modelo está construido con datos que cumplen las condiciones de distribución e independencia, entonces

ninguno de estos gráficos debería mostrar cambios sistemáticos en la distribución de los residuos en el eje  $y$  a lo largo del eje  $x$ . Por ejemplo, no deberían observarse tendencias no lineales, variación en las tendencias ni puntos aislados.

Para generar estos gráficos de residuos para el modelo ejemplo que se muestran en la figura 13.9, la llamada a la función `residualPlots()` en las líneas 13–15 del script 13.3, además de solicitar residuos en la escala estandarizada, indica que se usen puntos redondos rellenos (`pch = 20`) de color azul-acero (`col = "steelblue"`), marcando la línea de curvatura en color rojo (`col.quad = "red"`). También indica que se identifique a los tres residuos más alejados de la línea del cero (`id = list(method = "r", n = 3)`) usando texto con 70% del tamaño normal (`cex = 0.7`) a la izquierda o a la derecha (`location = "lr"`) del punto que lo representa en el gráfico.

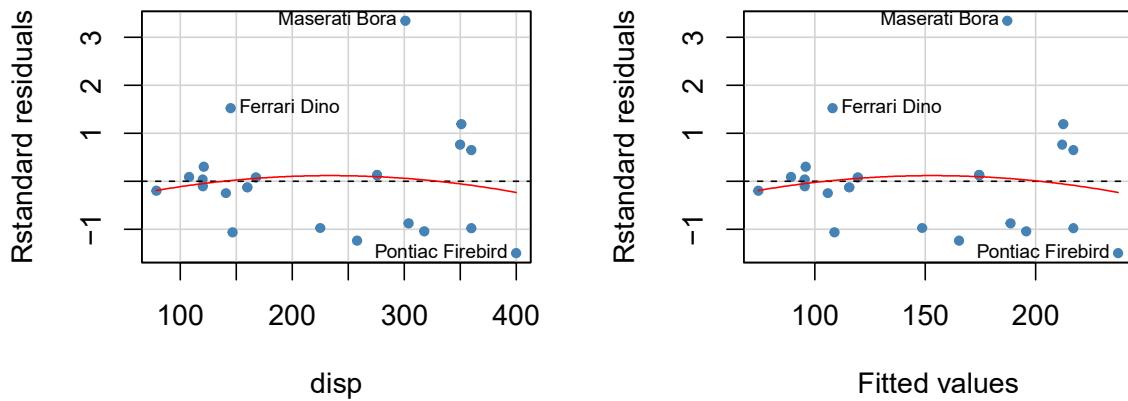


Figura 13.9: gráficos de los residuos estandarizados del modelo de RLS ejemplo.

La función `residualPlots()`, para conseguir las líneas de curvatura en los gráficos, realiza una prueba t de Student para comparar el ajuste del predictor con el ajuste conseguido al usar los valores del predictor al cuadrado. Para el gráfico de residuos en función de los valores ajustados, realiza la prueba de no aditividad de Tukey (Fox & Weisberg, 2018, cap. 6), agregando los cuadrados de los valores ajustados al modelo y volviendo a ajustar. Además de agregar las curvas al gráfico, la función despliega en pantalla el resultado de estas pruebas, que para el ejemplo resultan:

#### Pruebas de curvatura:

	Test stat	Pr(> Test stat )
disp	-0.378	0.7090
Tukey test	-0.378	0.7054

Vemos que las pruebas no resultan significativas. Tampoco se observan patrones sistemáticos en los gráficos de residuos (figura 13.9), aunque se aprecia la presencia de un valor atípico (el modelo Maserati Bora) que podría estar afectando la calidad del modelo.

Si bien no se ven patrones evidentes, aún cabe la posibilidad que exista alguna dependencia interna en los datos, menos evidente para el ojo humano pero significativa para la construcción de modelos de RLS, en la forma de **autocorrelación**, una característica típica de las series de tiempo.

El paquete `car` implementa la **prueba de Durbin-Watson** para detectar la presencia de autocorrelación en los residuos, una indicación clara de que las observaciones de la muestra no son independientes, sino que dependen de la observación anterior. Debemos notar que la distribución del estadístico de prueba  $DW$  utilizado en este procedimiento es compleja, y solo existen algunas tablas con unos cuantos valores conocidos. Por esta razón, la función `durbinWatsonTest(model)` implementada en este paquete obtiene un p-valor para el estadístico observado utilizando bootstrapping. Por esta razón, con el objetivo de asegurar reproducibilidad,

la línea 18 del script 13.3 fija una semilla antes de realizar la llamada a la función en la línea 19. El resultado obtenido es:

Prueba de independencia:

```
lag Autocorrelation D-W Statistic p-value
 1          0.3271828      1.341352   0.102
Alternative hypothesis: rho != 0
```

Así, fallamos en rechazar la hipótesis nula ( $DW = 1,341; p = 0,102$ ) y no podemos descartar que las observaciones que tenemos no presentan autocorrelación.

Otra función de diagnóstico útil del paquete `car` es `marginalModelPlots(modelo, sd = FALSE)`, donde:

- `modelo`: el modelo a emplear.
- `sd`: si tiene valor `TRUE`, marca líneas de desviación estándar estimadas en el gráfico para facilitar la evaluación de los supuestos sobre la varianza.

Esta función despliega gráficos, propuestos por Cook y Weisberg en 1997 (Fox & Weisberg, 2018, cap. 6), que tienen la variable de respuesta en el eje  $y$ , mientras que el eje  $x$  corresponde, sucesivamente, al predictor y a los valores ajustados. En el primer caso se muestra la distribución condicional entre la respuesta y el predictor. En el segundo caso, muestra la distribución condicional de la respuesta dada el ajuste del modelo. En cada uno, la función agrega, primero, una curva sólida que aproxima los valores observados de  $Y$  (por defecto se usa el método *lowess* (Cleveland, 1981) para obtener esta aproximación). Luego, agrega una curva punteada que aproxima las predicciones del modelo (los valores ajustados). Si el modelo es apropiado para los datos, entonces, ambas curvas deberían estimar el valor esperado condicional de la variable respuesta dado los valores del predictor  $y$ , y, en consecuencia, aproximadamente coincidir. Si esto no ocurre, tendríamos evidencia de que el modelo no se ajusta bien a los datos.

La figura 13.10 presenta los gráficos marginales para el modelo ejemplo, que se obtienen con la función `residualPlots()` llamada en las líneas 24–26 del script 13.3. Junto a indicar que se marquen estimaciones de las desviaciones estándar (`sd = TRUE`), se instruye que las aproximaciones basadas en los valores observados se dibuje en color azul-acero mientras que las aproximaciones basadas en los valores predichos se dibujen en rojo (`col.line = c("steelblue", "red")`). El resto de los argumentos son los mismos que fueron utilizados anteriormente en los gráficos de residuos para etiquetar los valores que generan los residuos más extremos.

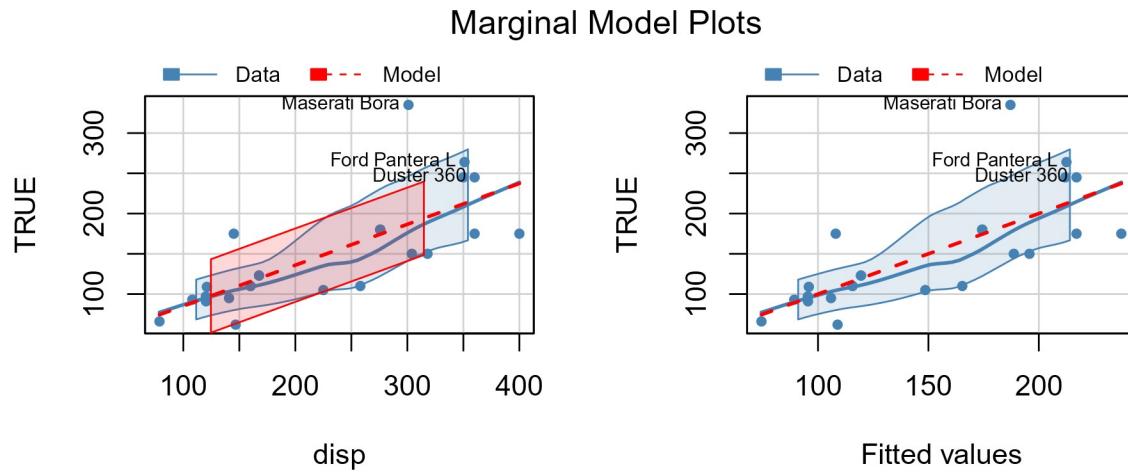


Figura 13.10: gráficos marginales para el modelo de RLS ejemplo.

En las figuras se puede observar algunas desviaciones entre las observaciones y las predicciones, en una región donde solo aparecen valores por debajo de la línea de regresión. Por otro lado, las estimaciones de las desviaciones estándar sugieren que la variabilidad va aumentando con el valor de la variable predictora.

Para verificar esta última sospecha, podemos usar la función `ncvTest(modelo)` del mismo paquete, que corresponde a una implementación de la prueba de varianza del error no constante (Fox & Weisberg, 2018, cap. 6). Para el ejemplo, aplicado en la línea 30 del script 13.3 , este resulta:

```
Prueba de homocedasticidad:
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 4.882653, Df = 1, p = 0.027128
```

Vemos que nos enfrentamos a un dilema: la condición de homocedasticidad parece no estarse cumpliendo con nivel  $\alpha = 0,05$  ( $\chi^2(1) = 4,883$ ;  $p = 0,027$ ). En estos casos debemos decidir un curso de acción. Puede que el modelo de todas formas refleje bien la relación entre las variables, y que la desviación se deba a las pocas observaciones que se tienen en la muestra. También, se podría intentar utilizar alguno de los métodos para hacer frente a datos problemáticos que vimos en capítulos anteriores. Es común que se usen técnicas para transformar los datos (como las llamadas *transformaciones estabilizadoras de la varianza*), aunque ya existen varios paquetes en R que implementan métodos robustos de regresión y funciones *wrapper* para aplicar remuestreo (de hecho, el paquete `car` proporciona algunas). Por último, queda desechar el modelo y buscar un mejor predictor.

Script 13.3: evaluación del modelo de regresión lineal simple usado como ejemplo.

```
1 library(car)
2 library(dplyr)
3 library(ggpubr)
4
5 # Cargar y filtrar los datos.
6 datos <- mtcars |> filter(wt > 2 & wt < 5)
7
8 # Ajustar modelo con R.
9 modelo <- lm(hp ~ disp, data = datos)
10
11 # Desplegar gráficos de residuos y mostrar pruebas de curvatura.
12 cat("Pruebas de curvatura:\n")
13 residualPlots(modelo, type = "rstandard",
14                 id = list(method = "r", n = 3, cex = 0.7, location = "lr"),
15                 col = "steelblue", pch = 20, col.quad = "red")
16
17 # Verificar independencia de los residuos
18 set.seed(19)
19 db <- durbinWatsonTest(modelo)
20 cat("\nPrueba de independencia:\n")
21 print(db)
22
23 # Desplegar gráficos marginales.
24 marginalModelPlots(modelo, sd = TRUE,
25                      id = list(method = "r", n = 3, cex = 0.7, location = "lr"),
26                      col = "steelblue", pch = 20, col.line = c("steelblue", "red"))
27
28 # Prueba de la varianza del error no constante.
29 cat("\nPrueba de homocedasticidad:\n")
30 print(ncvTest(modelo))
31
32 # Desplegar gráficos de influencia.
33 casos_influyentes <- influencePlot(modelo, id = list(cex = 0.7))
34 cat("\nCasos que podrían ser influyentes:\n")
35 print(casos_influyentes)
```

### 13.5.3 Influencia de los valores atípicos

Hemos dicho que los valores atípicos pueden ser síntomas del incumplimiento de las condiciones para usar RLS. También que puede ocurrir que la recta ajustada esté fuertemente influenciada por unos cuántos valores atípicos, por lo que el modelo se generaliza bien para la población. Sin embargo, no todos los valores atípicos son perjudiciales. La figura 13.11 muestra, para seis conjuntos de datos, los gráficos de dispersión (incluyendo la línea de regresión) y sus respectivos gráficos de los residuos. En cada uno de ellos se evidencia la presencia de al menos un valor atípico. Como señalan Diez et al. (2017, p. 349):

- En (1) hay un valor atípico que se aleja mucho de la nube de puntos, pero que no parece tener mucha influencia en la línea de regresión.
- En (2), se observa un valor atípico, a la derecha y bastante cercano a la línea de regresión, que no parece tener gran influencia.
- Nuevamente aparece un valor atípico a la derecha en (3), el cual parece ser el causante de que la línea de regresión no se ajuste muy bien a la nube principal de puntos.
- En (4), los datos se agrupan en dos nubes, una principal y otra secundaria con cuatro valores atípicos. La nube secundaria parece influenciar fuertemente la línea de regresión, haciendo que se ajuste pobremente a los datos de la nube principal.
- La nube principal no evidencia tendencia alguna (pendiente cercana a cero) en (5), pero el valor atípico a la derecha ejerce una gran influencia en la línea de regresión que se obtiene.
- En (6) se observa un valor atípico a la izquierda que se aleja bastante de la nube principal. Sin embargo, no parece ejercer mucha influencia en la línea de regresión y se sitúa cerca de ella.

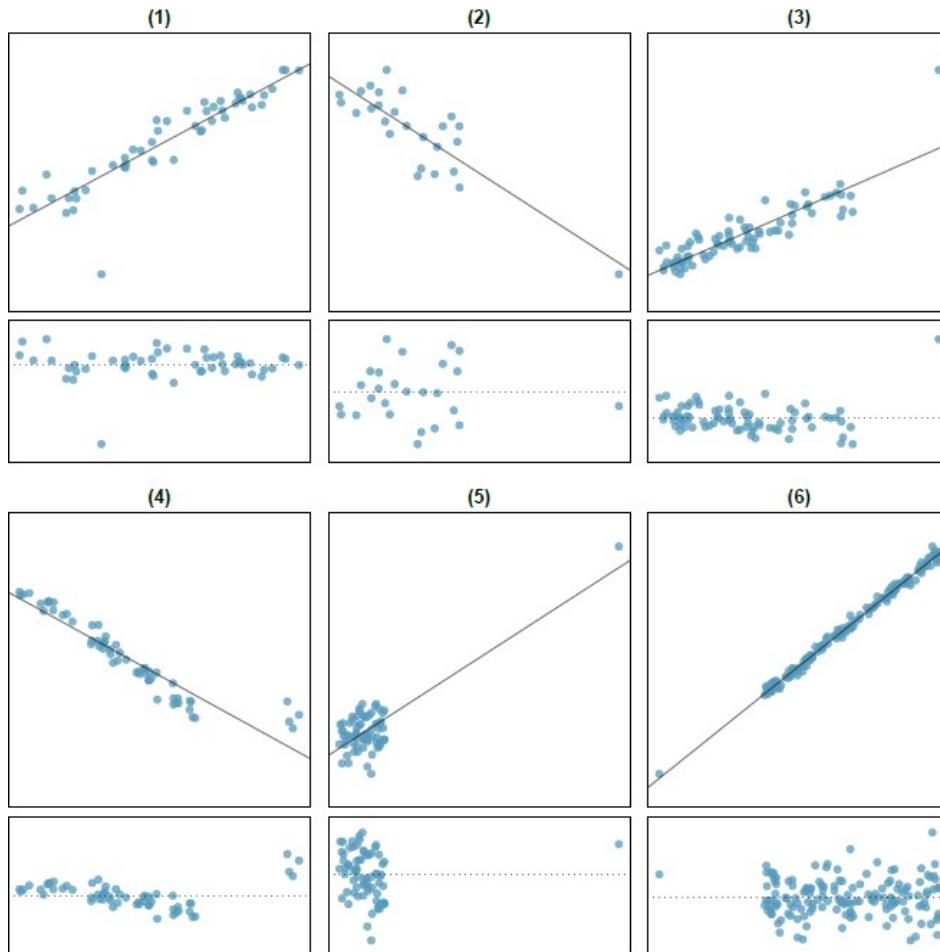


Figura 13.11: seis modelos de regresión lineal con sus respectivos gráficos de residuos. Fuente: Diez et al. (2017, p. 350).

Cuando un valor atípico ejerce un desplazamiento de la línea de regresión basada en la nube principal de puntos, como los encontrados en los gráficos (3), (4) y (5) de la figura 13.11, se le llama una **observación** o un **punto influyente**.

Los valores atípicos que se alejan horizontalmente del centro de la nube principal de puntos pueden, potencialmente, ejercer esta influencia. Por ello, un valor relacionado a qué tan lejos se encuentra un valor  $x_i$  respecto a la media muestral puede utilizarse como una medida del potencial de  $x_i$  para influir en el ajuste de la línea de regresión. A esta medida, se le conoce como **apalancamiento** (*leverage* en inglés), pues dichos puntos parecen “tirar” de la línea hacia ellos, aunque también es común que se refiera a ella como “valor *hat*”. Para el  $i$ -ésimo valor observado del predictor se calcula mediante la ecuación 13.9, donde:

- $n$  es el número de observaciones (tamaño de la muestra).
- $x_i, x_j$  son, respectivamente, el  $i$ -ésimo y  $j$ -ésimo valores observados del predictor.
- $\bar{x}$  es la media de todos los valores observados del predictor.

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^n (x_j - \bar{x})^2} \quad (13.9)$$

Como regla general, observaciones con valores de apalancamiento igual o mayores a 2 veces el valor promedio (que siempre corresponde a  $2/n$  para la RLS), son casos que podrían ser influyentes, pero no siempre indica un problema.

Otra forma, más intuitiva, de medir cuán influyente es una observación sería determinar cuánto cambia el modelo de regresión por su presencia. Esta idea está detrás de la **distorción de Cook** que mide el efecto combinado que tiene un punto sobre los coeficientes de regresión del modelo cuando es incluido o excluido del ajuste.

La ecuación 13.10 muestra cómo se calcula este estadístico para el  $i$ -ésimo punto, donde:

- $\hat{y}_j$  es la predicción para el  $j$ -ésimo valor observado del predictor usando el modelo completo (construido con toda la muestra).
- $\hat{y}_{j(-i)}$  es la predicción para el  $j$ -ésimo valor observado del predictor del modelo de RLS ajustado sin considerar la  $i$ -ésima observación.
- $k$  es el número de coeficientes en el modelo, igual a 2 para RLS.
- $MSE$  es el error cuadrático medio del modelo completo.

$$D_i = \frac{\sum_{j=1}^n (\hat{y}_j - \hat{y}_{j(-i)})^2}{k \ MSE} \quad (13.10)$$

Como regla general, un valor  $D_i > 1$  se considera que el punto es potencialmente influyente, aunque el valor  $4/n$  es usado como umbral en modelos construidos con muestras grandes, y observaciones que presenten valores mucho mayores al resto también son sospechosas.

El paquete `car` proporciona una función que permite revisar estas medidas de influencia en un solo gráfico tipo “burbuja” que muestra los residuos estandarizados en función de los valores de apalancamiento, donde el área de los círculos que representan las observaciones es proporcional a las distancias de Cook.

Las líneas 33–35 realizan la llamada a esta función, `influencePlot()`, cuyo resultado se presenta en la figura 13.12. Notemos las líneas punteadas verticales y horizontales que se observan en el gráfico. Las horizontales marcan los umbrales  $\pm 2$  en la escala de residuos estandarizados, que ayudan a reconocer residuos con valores atípicos. Las líneas de referencia verticales corresponden a dos y tres veces el valor promedio de apalancamiento, para identificar residuos con valores problemáticos.

Junto con desplegar este gráfico, la función retorna una matriz de datos con todos los casos que podrían estar ejerciendo una influencia problemática en el modelo, mostrada en pantalla en la línea 35 del script, resultando en:

Casos que podrían ser influyentes:

	StudRes	Hat	CookD
Fiat 128	-0.1933032	0.13622380	0.003075162
Pontiac Firebird	-1.5446806	0.16896761	0.228780764
Ferrari Dino	1.5702875	0.06961356	0.086721588
Maserati Bora	4.5756635	0.06337458	0.379425534

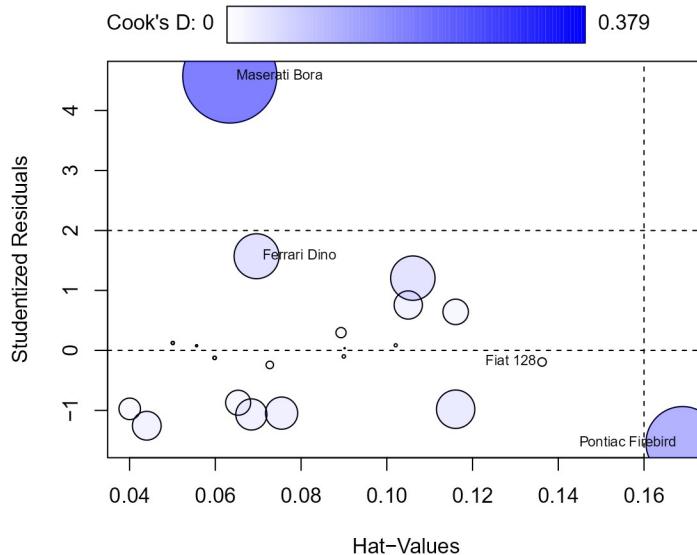


Figura 13.12: gráfico de diagnóstico para detectar posibles casos influyentes en el modelo del ejemplo.

Podemos ver que los modelos Pontiac Firebird y Maserati Bora muestran altos valores de la distancia de Cook y de apalancamiento o magnitud, respectivamente, por lo que podrían estar teniendo una influencia indebida en el modelo. Sin embargo, revisando los gráficos anteriores, esta posible sobreinfluencia no se observa con claridad.

Si bien puede resultar tentador descartar los valores que podrían ser problemáticos —o incluso, más radicalmente, eliminar los valores atípicos antes de ajustar el modelo—, no es pertinente llevar a cabo esta acción sin hacer un **riguroso análisis** previo. En muchos casos los valores atípicos resultan ser errores en el ingreso de los datos o rarezas que podemos ignorar, pero a veces estas resultan ser los casos más interesantes. Diez et al. (2017, p. 349) ilustran esta idea con el ejemplo de acciones de la bolsa con valores excepcionalmente altos. Si omitieran estos valores atípicos, los agentes de bolsa perderían los mejores negocios.

## 13.6 CALIDAD PREDICTIVA DE UN MODELO DE RLS

Hasta ahora hemos visto cómo evaluar si un modelo de RLS es confiable, es decir, que cumple las condiciones de distribución e independencia, no existen valores atípicos que alteren artificialmente la recta de regresión y que esta se ajusta bien a la muestra de datos.

Sin embargo, nos falta verificar si las predicciones que podemos obtener con él son de buena calidad.

### 13.6.1 Error de un modelo de RLS

En capítulos anteriores hemos discutido que, para una variable numérica  $Y$ , la calidad de un modelo puede asociarse a los errores  $e_i$  (positivos o negativos) que comete en las estimaciones  $\hat{y}_i$  de un conjunto de observaciones  $y_i$ , estableciendo la relación  $y_i = \hat{y}_i + e_i$ . Usando el modelo más simple, la relación puede escribirse

$y_i = \bar{y} + e_i$ . Cuando existe una variable categórica  $G$  que separa las observaciones en grupos, la relación puede tomar la forma  $y_i = \bar{y}_{|G=g} + e_i$ , donde  $g$  es el grupo al que pertenece la observación.

En este capítulo, al error que comete un modelo de RLS al estimar una observación  $i$  le hemos llamado el **residuo**  $r_i$ . Siguiendo la idea de arriba, podríamos hacer explícita la relación para este caso:  $y_i = \bar{y}_{|X=x_i} + r_i$ .

Así, como en casos anteriores podemos utilizar el **error cuadrático medio**, o MSE por sus siglas en inglés, como una métrica de la calidad de un modelo de RLS. Después de todo, esta es la medida que el método de mínimos cuadrados intenta minimizar. Desde luego, también podríamos usar la raíz cuadrada de este error, denotado RMSE, que se encuentra en la misma escala de la variable  $Y$ . Notemos que el MSE de un modelo de RLS corresponde a la suma de las desviaciones cuadradas entre los valores ajustados y las observaciones. Esto es la suma de los cuadrados de los residuos.

Para el ejemplo, el modelo consigue un  $\text{MSE} = 1.914,377$ ; equivalente a un  $\text{RMSE} = 43,754$ . En principio, este resultado podría ser satisfactorio considerando que la variable de salida tiene un rango que va de los 62 a los 335 caballos de fuerza, por lo que una predicción del modelo para un cierto automóvil, en promedio, no debería estar tan equivocada como para considerarlo entre los vehículos con motores de baja potencia cuando en realidad está entre los vehículos con motores de gran potencia, y viceversa. Sin embargo, para aplicaciones en que se requiera mayor precisión, el RMSE conseguido podría resultar inadecuado.

### 13.6.2 Generalización

Una pregunta obvia que aparece luego de conocer la calidad predictiva de un modelo de RLS, especialmente cuando el MSE resulta muy bajo, es si estos resultados serían similares para cualquier muestra de datos que pudiéramos conseguir. Diremos que un modelo es **generalizable** si para un conjunto de datos nuevo consigue predicciones con una calidad similar al que consigue con los datos usados en su construcción.

Por supuesto, al no contar con una bola de cristal mágica, solo podemos estimar la capacidad de generalización de un modelo. Si bien existe numerosas propuestas de como realizar esta estimación para modelos de RLS, estas también están sujetas a fuertes suposiciones sobre la distribución de las variables relacionadas. Por esta razón, mejor discutiremos métodos más genéricos que pueden aplicarse con cualquier tipo de modelo.

La estrategia más frecuente es la **validación cruzada** (CV en inglés, por *cross validation*), en la que el conjunto de datos se separa en dos fragmentos:

- **Conjunto de entrenamiento:** suele contener entre el 80 % y el 90 % de las observaciones (aunque es frecuente encontrar que solo contenga el 70 % de ellas), escogidas de manera aleatoria, y se emplea para ajustar la recta con el método de mínimos cuadrados.
- **Conjunto de prueba:** contiene el 10 % a 30 % restante de la muestra, y se usa para evaluar el modelo con datos nuevos.

Estos porcentajes se definen con el propósito de contar con la mayor cantidad de datos posible para ajustar el modelo, resguardando que el conjunto de prueba sea lo suficientemente grande como para obtener una buena estimación de la calidad del modelo. La idea detrás de este método es evaluar cómo se comporta el modelo con datos que no ha visto previamente, en comparación al comportamiento con el conjunto de entrenamiento.

El script 13.4 presenta la construcción y evaluación del modelo de RLS ejemplo usando CV. Como resultado, obtenemos el modelo descrito en la figura 13.13. Fijémonos en que, para el conjunto de entrenamiento, la raíz del error cuadrático medio es  $\text{RMSE}_e = 45,293$ , mientras que para el conjunto de prueba obtenemos  $\text{RMSE}_p = 45,288$ . Estos valores son muy parecidos, lo que sugiere que el modelo conseguido generaliza bien a otros datos.

Script 13.4: ajuste de una regresión lineal simple usando validación cruzada.

```

1 # Cargar y filtrar los datos.
2 datos <- mtcars |> filter(wt > 2 & wt < 5)
3 n <- nrow(datos)
4
5 # Crear conjuntos de entrenamiento y prueba.
6 set.seed(101)
```

```

Call:
lm(formula = hp ~ disp, data = entrenamiento)

Residuals:
    Min      1Q  Median      3Q     Max 
-48.96 -25.97 - 6.34 10.53 161.66 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 51.6585   28.5999   1.806 0.08763 .  
disp         0.4043    0.1211   3.339 0.00365 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 47.74 on 18 degrees of freedom
Multiple R-squared:  0.3825, Adjusted R-squared:  0.3482 
F-statistic: 11.15 on 1 and 18 DF,  p-value: 0.003653

MSE para el conjunto de entrenamiento: 45.29314
MSE para el conjunto de prueba: 45.2879

```

Figura 13.13: modelo de regresión lineal ajustado y evaluado usando validación cruzada.

```

7 n_entrenamiento <- floor(0.8 * n)
8 i_entrenamiento <- sample.int(n = n, size = n_entrenamiento, replace = FALSE)
9 entrenamiento <- datos[i_entrenamiento, ]
10 prueba <- datos[-i_entrenamiento, ]

11
12 # Ajustar y mostrar el modelo con el conjunto de entrenamiento.
13 modelo <- lm(hp ~ disp, data = entrenamiento)
14 print(summary(modelo))

15
16 # Calcular error cuadrado promedio para el conjunto de entrenamiento.
17 rmse_entrenamiento <- sqrt(mean(resid(modelo) ** 2))
18 cat("MSE para el conjunto de entrenamiento:", rmse_entrenamiento, "\n")

19
20 # Hacer predicciones para el conjunto de prueba.
21 predicciones <- predict(modelo, prueba)

22
23 # Calcular error cuadrado promedio para el conjunto de prueba.
24 error <- prueba[["hp"]] - predicciones
25 rmse_prueba <- sqrt(mean(error ** 2))
26 cat("MSE para el conjunto de prueba:", rmse_prueba)

```

Sin embargo, podría ocurrir que, por cosas del azar, el conjunto de prueba quedara compuesto por observaciones que no representan adecuadamente la muestra de datos original, llevando a una estimación equivocada de la calidad del modelo. Una manera de mejorar esta estimación es generalizar la estrategia anterior y usar **validación cruzada de k pliegues** (en inglés *k-fold cross validation*, abreviada k-CV). La idea sigue siendo usar un conjunto de entrenamiento para construir el modelo y otro de prueba para evaluarlo. Sin embargo, esta variante modifica este proceso a fin de obtener  $k$  estimaciones del MSE. Para ello se separa el conjunto de datos en  $k$  subconjuntos de igual tamaño y, como explica Amat Rodrigo (2016), obtenemos las estimaciones del error de la siguiente manera:

1. Para cada uno de los  $k$  subconjuntos:
  - a) Tomar uno de los  $k$  subconjuntos del conjunto de entrenamiento y reservarlo como conjunto de prueba.

- b) Ajustar la recta de mínimos cuadrados usando para ello los datos combinados de los  $k - 1$  subconjuntos restantes.
- c) Estimar el error cuadrático medio usando para ello el conjunto de prueba.
2. Estimar el error cuadrático medio del modelo, correspondiente a la media de los  $k$  MSE obtenidos en el paso 1.

El script 13.5 presenta la construcción del modelo de RLS ejemplo, esta vez, usando 5-CV. Para ello utilizamos la función `train(formula, method = "lm", trControl = trainControl(method = "cv", number))` del paquete `caret`, donde:

- `formula`: fórmula que se emplea en las llamadas internas a la función `lm()`.
- `number`: cantidad de pliegues ( $k$ ).

El lector atento notará que hemos asignado valores fijos a algunos de los argumentos de la función `train()`. Esto se debe a que este método sirve para ajustar muchos otros tipos de modelos, y estos valores son los que se necesitan para construir un modelo de RLS. También habrá observado que hemos fijado una semilla antes de llamar a esta función, puesto que, obviamente, realiza operaciones de muestreo aleatorio. El resultado del script se muestra en la figura 13.14.

```
Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max 
-62.382 -38.677   1.624   5.630 147.845 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 34.44423  23.47413   1.467   0.156    
disp         0.50734   0.09503   5.339 2.02e-05 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 45.62 on 23 degrees of freedom
Multiple R-squared:  0.5534, Adjusted R-squared:  0.534 
F-statistic: 28.5 on 1 and 23 DF,  p-value: 2.02e-05

Errores en cada pliegue:
        RMSE Rsquared       MAE Resample
1 15.16165 0.9893950 10.14825    Fold1
2 59.69373 0.6284084 51.17653    Fold2
3 71.25386 0.6151492 41.84016    Fold3
4 40.11558 0.5644574 29.96931    Fold4
5 38.67220 0.7585332 30.99186    Fold5

Error estimado para el modelo:
    intercept      RMSE Rsquared       MAE    RMSESD RsquaredSD      MAESD
1      TRUE 44.97941 0.7111886 32.82522 21.56035  0.1712064 15.36068
```

Figura 13.14: modelo de regresión lineal ajustado y evaluado usando validación cruzada de cinco pliegues.

Script 13.5: ajuste de una regresión lineal simple usando validación cruzada de cinco pliegues.

```
1 library(caret)
2 library(dplyr)
3
4 # Cargar y filtrar los datos.
5 datos <- mtcars |> filter(wt > 2 & wt < 5)
```

```

6 n <- nrow(datos)
7
8 # Ajustar y mostrar el modelo usando validación cruzada de 5 pliegues.
9 set.seed(111)
10 entrenamiento <- train(hp ~ disp, data = datos, method = "lm",
11                         trControl = trainControl(method = "cv", number = 5))
12 modelo <- entrenamiento[["finalModel"]]
13 print(summary(modelo))
14
15 # Mostrar los resultados de cada pliegue.
16 cat("Errores en cada pliegue:\n")
17 print(entrenamiento[["resample"]])
18
19 # Mostrar el resultado estimado para el modelo.
20 cat("\nError estimado para el modelo:\n")
21 print(entrenamiento[["results"]])

```

La línea 13 del script 13.5 muestra el modelo resultante en pantalla. La línea 17 hace lo propio con el detalle del error observado en cada pliegue, mientras que la línea 21 muestra el resultado global (medias y desviaciones estándar).

Estos resultados son más inquietantes. Si bien el error medio es similar a los anteriores ( $\text{RMSE} = 44,979$ ), muestra una alta dispersión ( $\text{RMSE}_{SD} = 21,560$ ). Vemos que en el pliegue 1 el error fue  $\text{RMSE} = 15,162$ , mientras que para el pliegue 3 alcanzó  $\text{RMSE} = 71,254$ . Esto sugiere que existe una o más observaciones influyentes, cuya presencia o ausencia en los pliegues producen grandes cambios en los coeficientes de la recta de regresión, por lo que sería conveniente investigar esta posibilidad.

Sin embargo, cuando la muestra disponible es pequeña, como en este caso, una buena alternativa es usar **validación cruzada dejando uno fuera** (*leave-one-out cross validation* en inglés, abreviada LOOCV). El esquema es el mismo que para validación cruzada con  $k$  pliegues, pero ahora usaremos tantos pliegues como observaciones tenga el conjunto de datos (es decir,  $k = n$ ). En otras palabras, se construyen  $n$  modelos distintos, dejando una observación distinta de la muestra de datos sin ser considerada. Luego, se puede determinar el error que comete cada uno de estos modelos en la observación que no fue incluida en su construcción. La media del cuadrado de estos residuos sería una estimación del MSE que cometería el modelo en datos no vistos al ajustar la recta de regresión.

El script 13.6 muestra cómo se construye el modelo de RLS ejemplo usando LOOCV, cuyo resultado puede verse en la figura 13.5.

La línea 17 del script despliega en pantalla las predicciones obtenidas en cada iteración, junto al valor observado usado como dato de prueba. La línea 21 hace lo propio con el error promedio del modelo. La salida de estas sentencias se puede ver en la figura 13.6. Podemos ver que este resultado, aunque similar a los anteriores, es un poco mayor.

Script 13.6: ajuste de una regresión lineal simple usando validación cruzada dejando uno fuera.

```

1 library(caret)
2 library(dplyr)
3
4 # Cargar y filtrar los datos.
5 datos <- mtcars |> filter(wt > 2 & wt < 5)
6 n <- nrow(datos)
7
8 # Ajustar y mostrar el modelo usando validación cruzada de 5 pliegues.
9 set.seed(111)
10 entrenamiento <- train(hp ~ disp, data = datos, method = "lm",
11                         trControl = trainControl(method = "LOOCV"))
12 modelo <- entrenamiento[["finalModel"]]
13 print(summary(modelo))
14
15 # Mostrar los errores.

```

```

Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max 
-62.382 -38.677   1.624   5.630 147.845 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 34.44423  23.47413   1.467   0.156    
disp         0.50734   0.09503   5.339 2.02e-05 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 45.62 on 23 degrees of freedom
Multiple R-squared:  0.5534, Adjusted R-squared:  0.534 
F-statistic: 28.5 on 1 and 23 DF,  p-value: 2.02e-05

```

Figura 13.15: modelo de regresión lineal ajustado usando validación cruzada dejando uno fuera.

```

16 cat("Predicciones en cada pliegue:\n")
17 print(entrenamiento[["pred"]])
18
19 # Mostrar el resultado estimado para el modelo.
20 cat("\nError estimado para el modelo:\n")
21 print(entrenamiento[["results"]])

```

Un aspecto importante a tener en cuenta es que la función `train()` ajusta el modelo final con la totalidad del conjunto de datos. Si bien esta función nos ha servido para estimar la generalización del error del modelo, si nos fijamos en las figuras 13.5, 13.14 y 13.5, nos podremos dar cuenta que en todos ellos obtuvimos el mismo modelo, con los mismos coeficientes.

## 13.7 EJERCICIOS PROPUESTOS

- [13.1] Elige dos variables numéricas del conjunto de datos `mtcars`, distintas a las usadas como ejemplo en este capítulo, para construir un modelo de RLS y evalúa su confiabilidad, calidad predictiva y confiabilidad.
- [13.2] Elige dos variables numéricas del conjunto de datos `Prestige` del paquete `carData`, para construir un modelo de RLS y evalúa su confiabilidad, calidad predictiva y confiabilidad.
- [13.3] Investiga qué es la métrica MAE que aparece en los reportes de error producidos por la función `train()`.
- [13.4] La función `train()`, ¿permite evaluar modelos de RLS usando bootstrapping?

## 13.8 BIBLIOGRAFÍA DEL CAPÍTULO

- Amat Rodrigo, J. (2016). *Validación de modelos predictivos: Cross-validation, OneLeaveOut, Bootstrapping*. Consultado el 23 de diciembre de 2021, desde [https://www.cienciadedatos.net/documentos/30\\_cross-validation\\_oneleaveout\\_bootstrap#K-Fold\\_Cross-Validation](https://www.cienciadedatos.net/documentos/30_cross-validation_oneleaveout_bootstrap#K-Fold_Cross-Validation)
- Cleveland, W. S. (1981). LOWESS: A program for smoothing scatterplots by robust locally weighted regression. *The American Statistician*, 35(1), 54.

Predicciones en cada pliegue:

		pred	obs	rowIndex	intercept
Mazda RX4		115.97706	110	1	TRUE
Mazda RX4 Wag		115.97706	110	2	TRUE
Datsun 710		88.80973	93	3	TRUE
Hornet 4 Drive		167.88673	110	4	TRUE
Hornet Sportabout		222.61469	175	5	TRUE
Valiant		150.41478	105	6	TRUE
Duster 360		213.42358	245	7	TRUE
Merc 240D		112.31355	62	8	TRUE
Merc 230		106.73128	95	9	TRUE
Merc 280		119.26762	123	10	TRUE
Merc 280C		119.26762	123	11	TRUE
Merc 450SE		174.07316	180	12	TRUE
Merc 450SL		174.07316	180	13	TRUE
Merc 450SLC		174.07316	180	14	TRUE
Fiat 128		75.69264	66	15	TRUE
Toyota Corona		95.21547	97	16	TRUE
Dodge Challenger		199.51630	150	17	TRUE
AMC Javelin		191.38024	150	18	TRUE
Camaro Z28		208.14497	245	19	TRUE
Pontiac Firebird		250.06592	175	20	TRUE
Porsche 914-2		95.92053	91	21	TRUE
Ford Pantera L		206.41402	264	22	TRUE
Ferrari Dino		102.99685	175	23	TRUE
Maserati Bora		177.15148	335	24	TRUE
Volvo 142E		94.54153	109	25	TRUE

Error estimado para el modelo:

	intercept	RMSE	Rquared	MAE
1	TRUE	47.52829	0.4769892	32.28154

Figura 13.16: errores del modelo de regresión lineal ajustado con validación cruzada dejando uno fuera.

- Diez, D., Barr, C. D., & Çetinkaya-Rundel, M. (2017). *OpenIntro Statistics* (3.<sup>a</sup> ed.).  
<https://www.openintro.org/book/os/>.
- Field, A., Miles, J., & Field, Z. (2012). *Discovering statistics using R*. SAGE Publications Ltd.
- Fox, J., & Weisberg, S. (2018). *An R companion to applied regression*. Sage publications.
- Glen, S. (2021). *Coefficient of Determination (R Squared)*. Consultado el 10 de junio de 2021, desde  
<https://www.statisticshowto.com/probability-and-statistics/coefficient-of-determination-r-squared/>
- Irizarry, R. A. (2019). *Introduction to Data Science*. <https://rafalab.github.io/dsbook/>.
- Pardoe, I., Simon, L., & Young, D. (2018). *Residuals vs. Fits Plot*.  
 Consultado el 21 de diciembre de 2021, desde <https://online.stat.psu.edu/stat462/node/117/>
- Winner, L. (2021). *Simple Linear Regression I — Least Squares Estimation*.  
 Consultado el 8 de junio de 2021, desde <http://users.stat.ufl.edu/~winner/qmb3250/notespart2.pdf>