



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 y col. (2017, pp. 331-355), Field y col. (2012, pp. 245-311) e Irizarry (2019, pp. 535-545)

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

- β_0 y β_1 son los parámetros del modelo lineal.
- x es la variable explicativa o **predictor** (variable independiente).
- y es 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 β_0 , que corresponde al punto en que la recta corta el eje y . A su vez, denominamos **pendiente** al parámetro β_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.

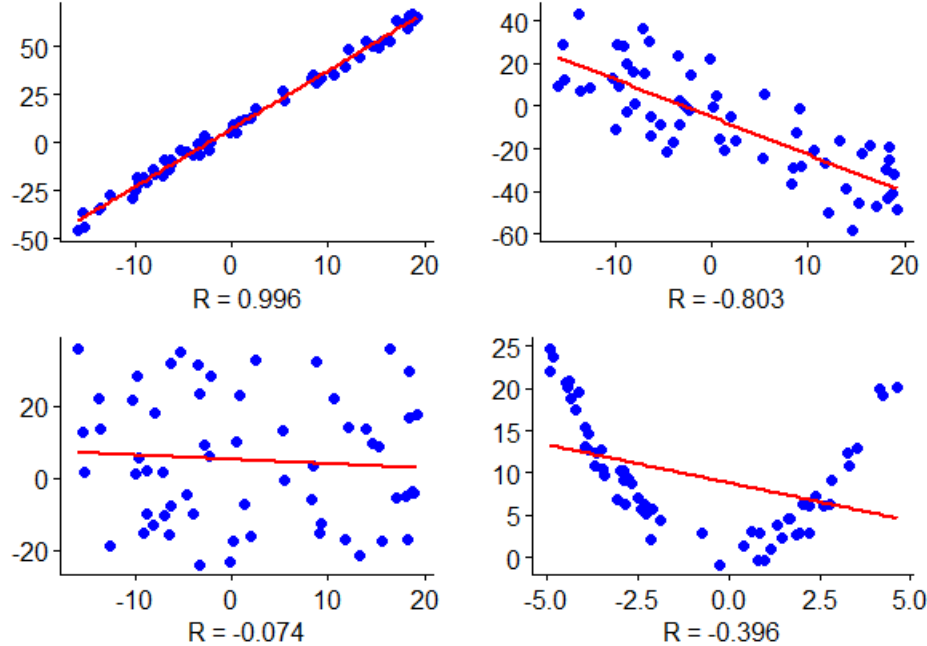


Figura 13.1: modelos lineales para cuatro conjuntos de datos.

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 aglutinan 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 . 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 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.

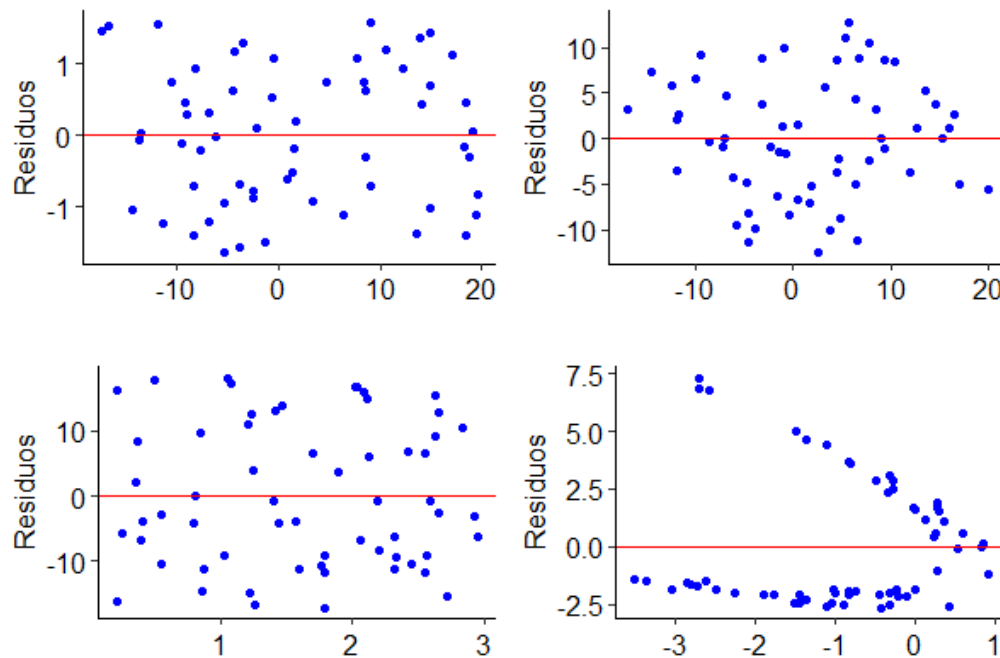


Figura 13.2: residuos para los modelos lineales de la figura 13.1.

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 mediante la **correlación**. Una de las formas más sencillas para calcularla es el coeficiente de correlación de Pearson, dado por la ecuación 13.3, 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 de las variables x e y en la muestra.
- n es el tamaño de la muestra.

$$R = \frac{1}{n-1} \cdot \sum_{i=1}^n \frac{x_i - \bar{x}}{s_x} \cdot \frac{y_i - \bar{y}}{s_y} \quad (13.3)$$

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 ($R < 0$). Para comprender mejor esta idea, fijémonos los coeficientes de correlación obtenidos para cada modelo lineal de la figura 13.1, indicados en las etiquetas del eje x . Podemos ver que, en el caso 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 los ejemplos de este capítulo usaremos 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 [millas/galón].
cyl	Cantidad de cilindros del motor.
disp	Volumen útil de los cilindros de un motor, en centímetros cúbicos [cc].
hp	Potencia del motor, en caballos de fuerza [hp].
drat	Relación del eje trasero (proporción).
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.
carb	Número de carburadores.

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

Si consideramos a x como el rendimiento del vehículo y a y como la potencia del motor, cuya relación se muestra gráficamente en la figura 13.3, tenemos que: $\bar{x} = 20,091$, $s_x = 6,027$, $\bar{y} = 146,688$ y $s_y = 68,563$. En consecuencia, la correlación es:

$$R = \frac{1}{32-1} \cdot \sum_{i=1}^n \frac{x_i - 20,091}{6,027} \cdot \frac{y_i - 146,688}{68,563} = -0,776$$

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 obtenemos que $R = -0,7761684$, lo que coincide con el resultado teórico 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 todos los pares de variables. La figura 13.4 muestra la matriz de correlación

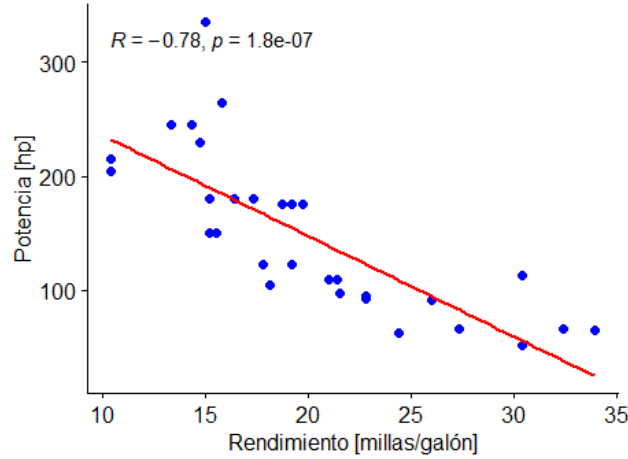


Figura 13.3: Relación entre el rendimiento y la potencia.

(redondeada al segundo decimal) para el conjunto de datos `mtcars`. Podemos ver que, naturalmente, la matriz de correlación es simétrica y que su diagonal solo contiene unos.

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
mpg	1.00	-0.85	-0.85	-0.78	0.68	-0.87	0.42	0.66	0.60	0.48	-0.55
cyl	-0.85	1.00	0.90	0.83	-0.70	0.78	-0.59	-0.81	-0.52	-0.49	0.53
disp	-0.85	0.90	1.00	0.79	-0.71	0.89	-0.43	-0.71	-0.59	-0.56	0.39
hp	-0.78	0.83	0.79	1.00	-0.45	0.66	-0.71	-0.72	-0.24	-0.13	0.75
drat	0.68	-0.70	-0.71	-0.45	1.00	-0.71	0.09	0.44	0.71	0.70	-0.09
wt	-0.87	0.78	0.89	0.66	-0.71	1.00	-0.17	-0.55	-0.69	-0.58	0.43
qsec	0.42	-0.59	-0.43	-0.71	0.09	-0.17	1.00	0.74	-0.23	-0.21	-0.66
vs	0.66	-0.81	-0.71	-0.72	0.44	-0.55	0.74	1.00	0.17	0.21	-0.57
am	0.60	-0.52	-0.59	-0.24	0.71	-0.69	-0.23	0.17	1.00	0.79	0.06
gear	0.48	-0.49	-0.56	-0.13	0.70	-0.58	-0.21	0.21	0.79	1.00	0.27
carb	-0.55	0.53	0.39	0.75	-0.09	0.43	-0.66	-0.57	0.06	0.27	1.00

Figura 13.4: matriz de correlación para el conjunto de datos `mtcars`

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 13.4).

$$\min \sum_{i=1}^n e_i^2 \quad (13.4)$$

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 y col. (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. Los datos deben presentar una relación lineal.
2. La distribución de los residuos debe ser cercana a la normal.
3. La variabilidad de los puntos en torno a la línea de mínimos cuadrados debe ser aproximadamente constante.
4. Las observaciones deben ser 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).

Los gráficos de residuos reflejan cuando no se cumplen las condiciones anteriores. Por ejemplo, el gráfico inferior derecho de la figura 13.1 aplica regresión lineal entre un par de variables cuya relación es, en realidad, cuadrática. Esta relación se puede ver en la forma en que se distribuyen los puntos en el gráfico de residuos correspondiente de la figura 13.2.

La figura 13.5 muestra, en la fila superior, dos modelos lineales en los que los datos no cumplen las condiciones, con sus respectivos gráficos de residuos en la fila inferior. A la izquierda, se viola la condición de normalidad de los residuos, que no se distribuyen aleatoriamente en torno a la línea 0. A la derecha, no se respeta la condición de homocedasticidad, y la variabilidad de los puntos en torno a la línea de mínimos cuadrados no es aproximadamente constante, lo que genera una característica forma de embudo en el gráfico de residuos.

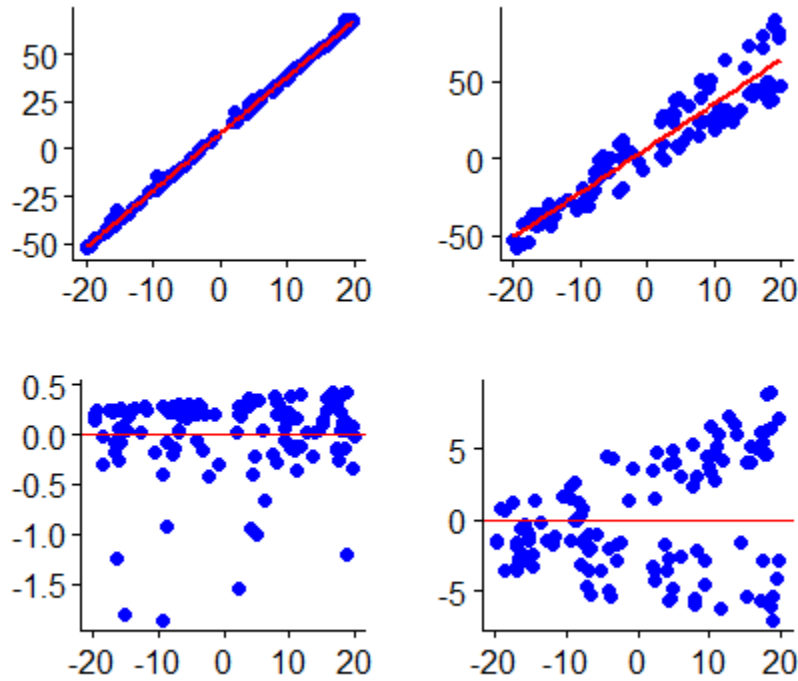


Figura 13.5: modelos lineales (fila superior) que violan alguna condición y sus residuos (fila inferior).

El primer paso que debemos seguir cuando queremos determinar la recta de mínimos cuadrados para un conjunto de datos consiste en estimar la pendiente (β_1) mediante la ecuación 13.5, 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.

$$b_1 = \frac{s_y}{s_x} \cdot R \quad (13.5)$$

El punto (\bar{x}, \bar{y}) , donde \bar{x} e \bar{y} son las medias muestrales para las variables representadas en los ejes x e y respectivamente, siempre pertenece a la recta de mínimos cuadrados, por lo que podemos calcular la intercepción mediante la ecuación 13.6 (Winner, 2021, p. 4).

$$b_0 = \bar{y} - b_1 \cdot \bar{x} \quad (13.6)$$

Cuando contamos con más de una variable para construir una regresión lineal simple (RLS), lo más adecuado es que escojamos como predictor aquella variable que tenga la correlación más fuerte con la variable de respuesta. Para el caso del conjunto de datos `mtcars`, la variable que presenta la correlación más fuerte con el rendimiento (`mpg`) es el peso del automóvil (`wt`), con $R = -0,87$, como podemos ver en la figura 13.4. Así, si usamos como predictor la variable peso, tenemos que la pendiente de la recta es:

$$b_1 = \frac{6,027}{0,978} \cdot -0,868 = -5,344$$

A su vez, la intercepción está dada por:

$$b_0 = 20,091 + 5,344 \cdot 3,217 = 37,285$$

Por lo que la recta ajustada mediante mínimos cuadrados es:

$$\widehat{\text{mpg}} = 37,285 - 5,344 \cdot \text{wt}$$

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 rendimiento (`mpg`), con la variable peso (`wt`) como predictor, mediante el uso de `lm()`. En la figura 13.6, bajo el encabezado **Coefficients**, podemos ver que los valores estimados para los parámetros de la RLS coinciden con los obtenidos previamente.

Un detalle interesante es que, en la línea 8 del script 13.1, usamos la llamada `print(summary(modelo))` en lugar de `print(modelo)`, lo que nos entrega información más detallada del modelo ajustado (figura 13.6). La segunda solo muestra los coeficientes obtenidos.

```
Call:
lm(formula = mpg ~ wt, data = datos)

Residuals:
    Min       1Q   Median       3Q      Max
-4.5432 -2.3647 -0.1252  1.4096  6.8727

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.2851     1.8776  19.858 < 2e-16 ***
wt          -5.3445     0.5591  -9.559 1.29e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
F-statistic: 91.38 on 1 and 30 DF, p-value: 1.294e-10
```

Figura 13.6: regresión lineal simple para predecir el rendimiento de un automóvil a partir de su peso.

La figura 13.7 muestra la recta ajustada para el rendimiento de un automóvil de acuerdo a su peso (script 13.1, líneas 11–15), donde podemos observar que los datos presentan una relación lineal, aunque algunos puntos parecen estar algo alejados de la recta ajustada. A su vez, la figura 13.8 muestra diversos gráficos obtenidos mediante la línea 18 del script 13.1, donde vemos que la variabilidad de los residuos no es muy grande (figura 13.8a) y, en general, sigue una distribución razonablemente cercana a la normal (figura 13.8b), aunque en ambas figuras se aprecian unos pocos modelos que se comportan como valores atípicos. Además, podemos suponer que las observaciones son independientes entre sí y, evidentemente, no corresponden a una serie de tiempo. Con este análisis verificamos las condiciones 1 y 4 para emplear la regresión lineal de mínimos cuadrados, aunque las dos condiciones restantes parecen no cumplirse.

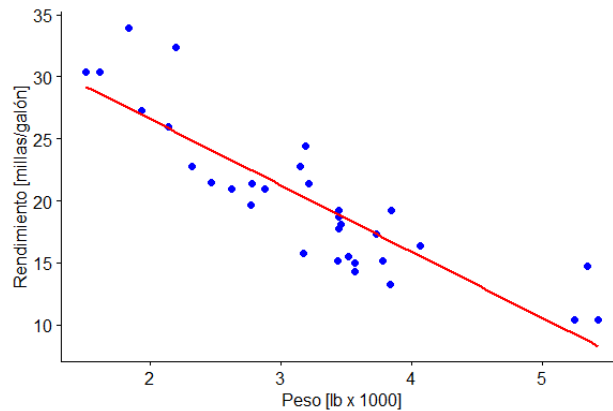


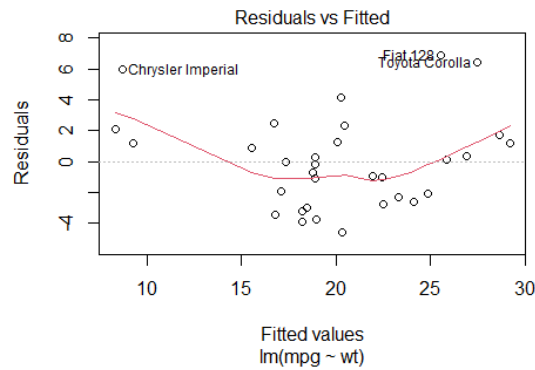
Figura 13.7: recta ajustada para el rendimiento de un automóvil de acuerdo a su peso.

Si bien para fines del ejercicio supondremos que las condiciones se cumplen, vale la pena que revisemos algunas características que, según Pardoe y col. (2018), se observan en el gráfico de los residuos cuando sí se verifican todas las condiciones o, en otras palabras, cuando el modelo de RLS es apropiado:

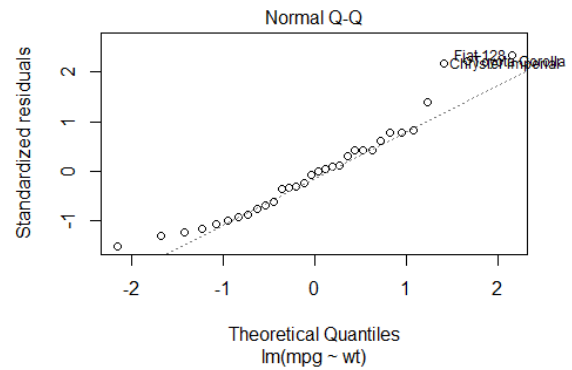
1. Un gráfico en que los residuos se distribuyen aleatoriamente en torno a la línea de valor 0, sugiere que es razonable suponer que las variables presentan una relación lineal.
2. Cuando los residuos forman una “banda horizontal” en torno a la línea de valor 0, sugiere una variabilidad aproximadamente constante de los residuos.
3. La ausencia de residuos que se alejen del patrón que forman los demás sugiere la ausencia de valores atípicos.

Script 13.1: ajuste de una regresión lineal simple.

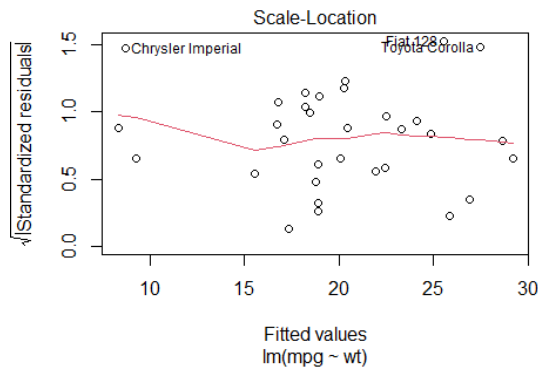
```
1 library(ggpubr)
2
3 # Cargar los datos.
4 datos <- mtcars
5
6 # Ajustar modelo con R.
7 modelo <- lm(mpg ~ wt, data = datos)
8 print(summary(modelo))
9
10 # Graficar el modelo.
11 p <- ggscatter(datos, x = "wt", y = "mpg", color = "blue", fill = "blue",
12               xlab = "Peso [lb x 1000]", ylab = "Rendimiento [millas/galón]")
13
14 p <- p + geom_smooth(method = lm, se = FALSE, colour = "red")
15 print(p)
16
17 # Crear gráficos para evaluar el modelo.
18 plot(modelo)
19
```



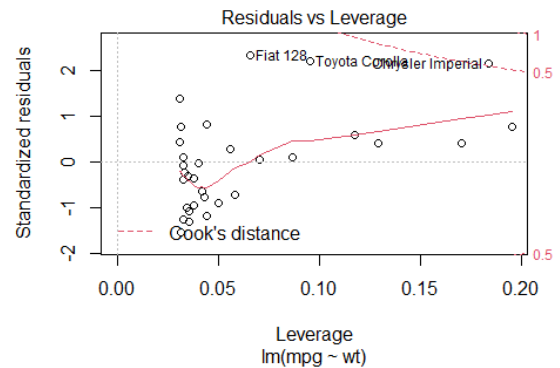
(a) residuos.



(b) distribución de los residuos.



(c) residuos estandarizados.



(d) apalancamiento.

Figura 13.8: gráficos para evaluar el modelo lineal.

```

20 # Ingresar algunas instancias artificiales.
21 mpg <- c(23.714, 19.691, 19.242, 12.430, 10.090, 9.565, 18.171, 26.492, 7.054,
22         24.447, 15.683, 17.403, 13.465, 18.850, 29.493)
23
24 wt <- c(2.973, 4.532, 2.332, 3.016, 4.220, 4.286, 2.580, 3.084, 3.816, 2.775,
25        3.251, 3.013, 4.951, 2.644, 2.218)
26
27 nuevos <- data.frame(mpg, wt)
28
29 # Usar el modelo para predecir el rendimiento de los nuevos y ver los
30 # residuos resultantes.
31 predicciones <- predict(modelo, nuevos)
32 residuos <- nuevos$mpg - predicciones
33 nuevos <- data.frame(nuevos, residuos)
34
35 r <- ggscatter( nuevos, x = "wt", y = "residuos", color = "blue",
36               fill = "blue", xlab = "Peso [lb * 1000]", ylab = "Residuo")
37
38 r <- r + geom_hline(yintercept = 0, colour = "red")
39 print(r)

```

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 respuesta y si el predictor x se incrementa

en una unidad. Así, para el ejemplo se espera que, al incrementar en el peso del automóvil en 1.000 libras, el rendimiento se reduzca en 5,344 millas por galón de combustible. 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 el ejemplo, es imposible que un automóvil carezca de masa.

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 peso de los automóviles del ejemplo varía entre 1.500 y 5.500 libras aproximadamente, por lo que si lo usáramos para predecir el rendimiento de un vehículo de 7.000 libras o de solo 980, el resultado podría carecer de validez.

Desde luego, debemos tener en cuenta también las condiciones de diseño del modelo. El resultado podría ser equivocado si intentáramos predecir, por ejemplo, el rendimiento de un automóvil moderno (recordemos que el conjunto de datos solo contiene vehículos de los años 1973 y 1974). Más aún, la variable de respuesta carece absolutamente de sentido si pensamos, por ejemplo, en un automóvil eléctrico.

13.3 USO DEL MODELO

Supongamos que queremos predecir el rendimiento de un auto norteamericano (modelo 1974) cuyo peso es de 4.260 libras (es decir, `wt = 4,260`). Para ello, basta con reemplazar el valor del predictor en el modelo:

$$\widehat{\text{mpg}} = 37,285 - 5,344 \cdot 4,260 = 14,520$$

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

- **object**: el modelo a emplear.
- **newdata**: matriz de datos con las nuevas instancias para las que se desea efectuar la predicción, la cual debe tener todas las columnas presentes en la fórmula del modelo (para el ejemplo, `mpg` y `wt`).

La línea 31 del script 13.1 ilustra el uso de esta función para un conjunto de 15 instancias generadas artificialmente.

En la línea 32 se calculan los residuos para posteriormente graficarlos (líneas 35–39) a fin de tener una idea preliminar acerca de la calidad de las predicciones. Como resultado obtenemos la figura 13.9, donde podemos observar que los residuos varían en un rango bastante más amplio que para el conjunto de datos original (figura 13.8a).

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

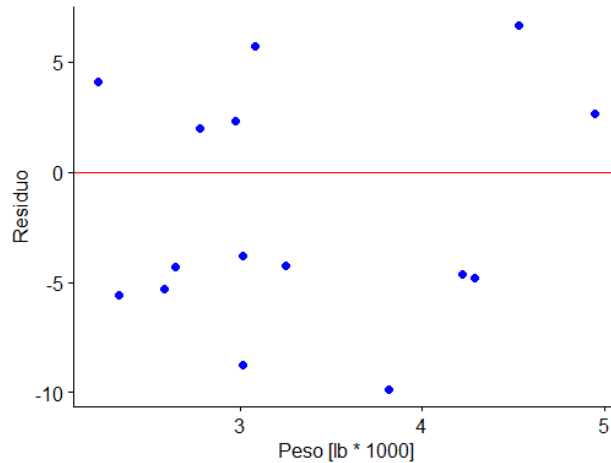


Figura 13.9: residuos obtenidos tras usar el modelo para predecir el rendimiento de nuevos automóviles.

sencillo, como muestra el script 13.2. En la práctica, rara vez tendremos que realizar este paso, pues las funciones de R que ajustan modelos lo hacen automáticamente cuando encuentran predictores categóricos.

Script 13.2: reemplazar una variable dicotómica por una variable indicadora.

```
1 # Crear un data frame con una variable dicotómica.
2 alumno <- 1:5
3 sexo <- factor(c("F", "M", "F", "F", "M"))
4 datos <- data.frame(alumno, sexo)
5
6 # Crear una variable indicadora para sexo, con valor 0
7 # para hombres y 1, para mujeres.
8 es_mujer <- rep(1, length(sexo))
9 es_mujer[sexo == "M"] <- 0
10
11 # Reemplazar la variable sexo por la variable indicadora.
12 datos <- cbind(datos, es_mujer)
13 datos[["sexo"]] <- NULL
```

El conjunto de datos **mtcars** ya cuenta con un par de variables que cumplen con esta característica: la transmisión (**am**) y la forma del motor (**vs**). De estas dos variables, la forma del motor tiene una correlación más fuerte con el rendimiento, por lo que la usaremos como ejemplo para crear un modelo RLS. Al crear el modelo (script 13.3) obtenemos como resultado la recta representada en el gráfico superior de la figura 13.10, con los residuos del gráfico inferior en la misma figura. A su vez, la figura 13.11 muestra los valores obtenidos para los parámetros del modelo.

Cuando usamos un predictor dicotómico siempre se cumple la condición de que los datos presentan una relación lineal. Sin embargo, debemos verificar que la distribución de los residuos de ambos grupos se asemeje a la normal y que tengan varianzas similares. El panel superior de la figura 13.10 muestra que, en efecto, las variabilidades de los residuos de ambos grupos son independientes y la figura 13.12 muestra que la distribución de los residuos se acerca a la normal para ambos tipos de transmisión, por lo que se verifican las condiciones.

Script 13.3: alternativa robusta para comparar entre múltiples grupos correlacionados.

```
1 library(ggpubr)
2
3 # Cargar los datos.
4 datos <- mtcars
5
6 # Ajustar modelo con R.
```

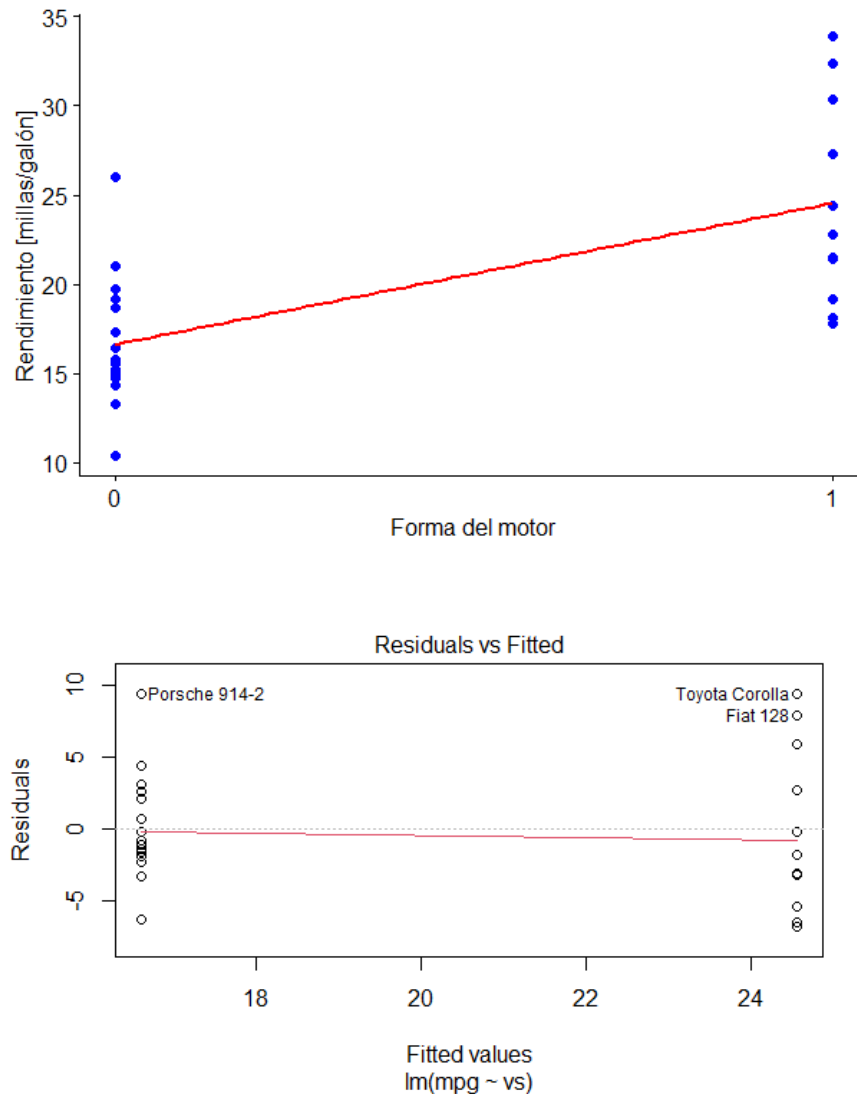


Figura 13.10: modelo de regresión lineal y gráfico de residuos para el ejemplo con un predictor dicotómico.

```

7 modelo <- lm(mpg ~ vs, data = datos)
8 print(summary(modelo))
9
10 # Graficar el modelo.
11 p <- ggscatter(datos, x = "vs", y = "mpg", color = "blue", fill = "blue",
12               xlab = "Forma del motor", ylab = "Rendimiento [millas/galón]",
13               xticks.by = 1)
14
15 p <- p + geom_smooth(method = lm, se = FALSE, colour = "red")
16 print(p)
17
18 # Crear gráficos para evaluar el modelo.
19 plot(modelo)
20
21 #Graficar residuos.
22 residuos <- modelo$residuals
23 datos <- cbind(datos, residuos)
24 datos[["vs"]] <- factor(datos[["vs"]])

```

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

Residuals:
    Min       1Q   Median       3Q      Max
-6.757 -3.082 -1.267  2.828  9.383

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   16.617     1.080   15.390 8.85e-16 ***
vs             7.940     1.632    4.864 3.42e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.581 on 30 degrees of freedom
Multiple R-squared:  0.4409, Adjusted R-squared:  0.4223
F-statistic: 23.66 on 1 and 30 DF, p-value: 3.416e-05
```

Figura 13.11: recta de mínimos cuadrados para el ejemplo con un predictor dicotómico.

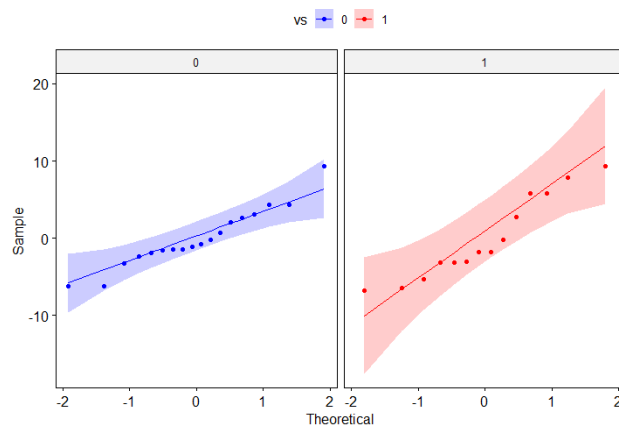


Figura 13.12: distribución de los residuos.

```
25
26 r <- ggqqplot(datos, x = "residuos", facet.by = "vs", color = "vs",
27               palette = c("blue", "red"))
28
29 print(r)
```

13.5 EVALUACIÓN DE UN MODELO DE RLS

Hasta ahora hemos visto cómo ajustar y usar un modelo de RLS. Sin embargo, no hemos realizado algunas verificaciones importantes antes de hacer predicciones, pues puede ocurrir que la recta ajustada esté fuertemente influenciada por un pequeño grupo de valores atípicos o que no pueda generalizarse para otras muestras.

13.5.1 Influencia de los valores atípicos

Hemos dicho que, en muchas ocasiones, los valores atípicos influyen significativamente en el incumplimiento de las condiciones que debemos verificar para poder usar una regresión lineal. Sin embargo, no todos los valores atípicos son perjudiciales. La figura 13.13 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 y col. (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 la 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), y el valor atípico a la derecha parece ejercer una gran influencia en la línea de regresión.
- 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.

Los valores atípicos que se alejan horizontalmente del centro de la nube principal de puntos pueden, potencialmente, tener una gran influencia en el ajuste de la línea de regresión. Este fenómeno se conoce como **apalancamiento** (*leverage* en inglés), pues dichos puntos parecen tirar de la línea hacia ellos. Cuando un valor atípico ejerce efectivamente esta influencia, decimos que es un **punto influyente**. Una forma de saber si un punto es o no influyente es determinar la línea de regresión sin considerar dicho punto y ver cuánto se aleja este último de la nueva línea. Si miramos la figura 13.8d, podemos detectar dos observaciones atípicas que influyen en el ajuste del modelo.

Si bien puede resultar tentador descartar los valores atípicos antes de ajustar un modelo, no es pertinente llevar a cabo esta acción sin hacer un riguroso análisis previo. En muchos casos los valores atípicos resultan ser las observaciones más interesantes. Diez y col. (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.

Un buen método para identificar valores atípicos es usar los residuos estandarizados (figura 13.8c) (es decir, divididos por la estimación de su desviación estándar), pues esto nos permite establecer un rango fijo de valores aceptables y, en consecuencia, fijar un criterio para comparar residuos de distintos modelos.

Debemos ser cuidadosos cuando usemos como predictores variables categóricas que tengan pocas observaciones en alguno de sus niveles, pues cuando esto ocurre, dichas observaciones se convierten en puntos influyentes.

13.5.2 Bondad de ajuste

Una medida muy útil que podemos usar para evaluar la **bondad de ajuste** de un modelo de regresión lineal con respecto a las observaciones es el **coeficiente de determinación**, que corresponde al cuadrado de la correlación, por lo que suele también denominarse R-cuadrado (R^2) (Glen, 2021). Esta medida, cuyo valor varía entre 0 y 1, corresponde al porcentaje de la variabilidad de la respuesta que es explicado por el predictor, dado por la ecuación 13.7, donde s_e^2 es la varianza de los residuos.

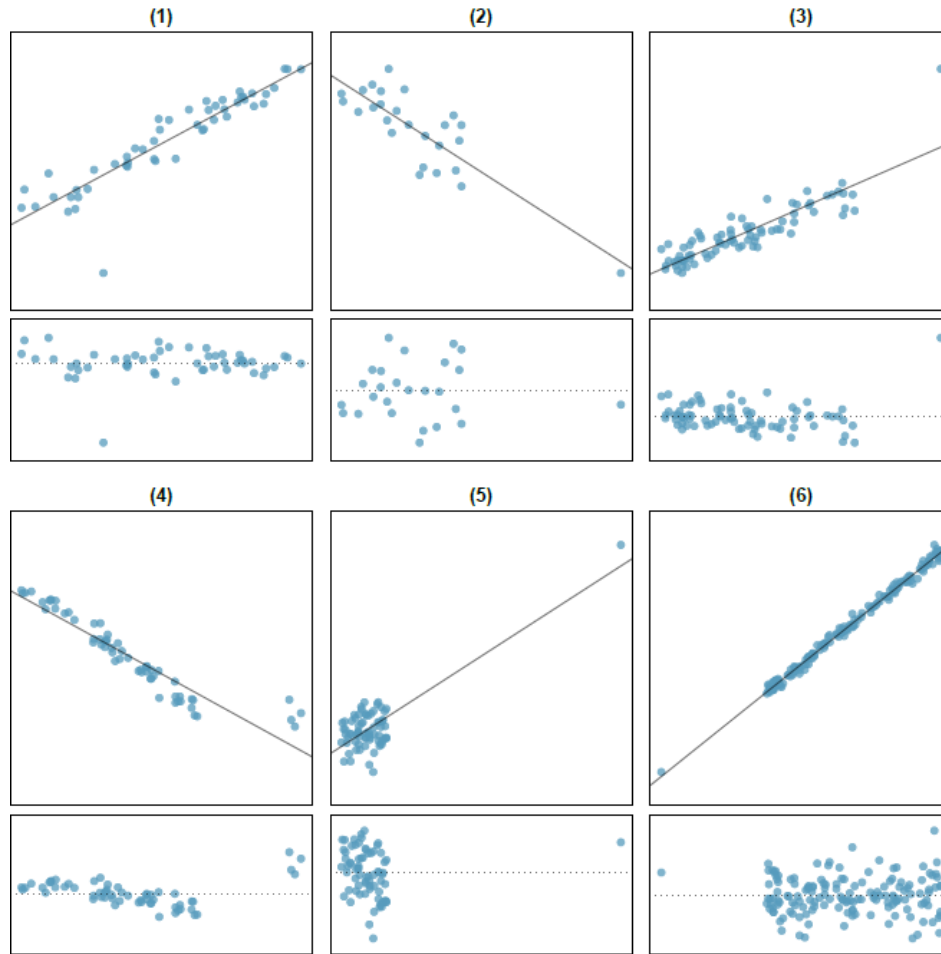


Figura 13.13: seis modelos de regresión lineal con sus respectivos gráficos de residuos. Fuente: Diez y col. (2017, p. 350).

$$R^2 = \frac{s_y^2 - s_e^2}{s_y^2} \quad (13.7)$$

Para el ejemplo, como podemos verificar en la penúltima línea de la descripción del modelo obtenido (figura 13.6) bajo el nombre de **Multiple R-squared**, tenemos:

$$R^2 = -0,868^2 = 0,753$$

En consecuencia, la recta de regresión lineal, construida con el peso del vehículo como predictor, explica 75,3 % de la variabilidad en el rendimiento.

13.5.3 Validación cruzada

Hasta ahora hemos visto cómo detectar valores atípicos que influyan en la recta y cómo determinar si la recta se ajusta bien a la muestra. Sin embargo, nos falta verificar si el modelo puede generalizarse. Una estrategia frecuente para esto es la **validación cruzada**, en la que el conjunto de datos se separa en dos fragmentos:

- **Conjunto de entrenamiento:** contiene entre el 80 % y el 90 % de las observaciones, escogidas de manera aleatoria, y se emplea para ajustar la recta con el método de mínimos cuadrados.
- **Conjunto de prueba:** contiene el 10 % a 20 % restante de las instancias, 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. Una buena métrica que podemos usar para esta tarea es el **error cuadrático medio**, o MSE por sus siglas en inglés, pues es lo que el método de mínimos cuadrados busca minimizar.

El script 13.4 aborda, una vez más, el ajuste de una RLS para predecir el rendimiento de un automóvil a partir de su peso, pero esta vez usando validación cruzada. Como resultado, obtenemos el modelo de la figura 13.14. Fijémonos en que, para el conjunto de entrenamiento, el error cuadrático medio es $MSE_e = 5,652$, mientras que para el conjunto de prueba obtenemos $MSE_p = 17,516$, bastante más elevado (¡más del triple!). Esto sugiere que el modelo puede estar sobreajustado, es decir, que se adapta bien a los datos del conjunto de entrenamiento pero no tanto al conjunto de prueba, por lo que podría ser imprudente suponer que puede ser generalizado. Sin embargo, esto puede deberse a la separación aleatoria de los datos. Al ejecutar el script 13.4 reemplazando la semilla aleatoria por 125, obtenemos el resultado de la figura 13.15. Podemos notar que los parámetros del modelo son algo diferentes a los obtenidos con la semilla 101. Además, ahora el error cuadrático medio para el conjunto de entrenamiento es $MSE_e = 8,596$ y para el conjunto de prueba, $MSE_p = 9,122$. Estos últimos valores son muy parecidos, por lo que este segundo modelo sí podría ser generalizable.

```
Call:
lm(formula = mpg ~ wt, data = entrenamiento)

Residuals:
    Min       1Q   Median       3Q      Max
-3.602 -1.854 -0.212  1.590  4.684

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  34.4745     2.1780  15.829 8.89e-13 ***
wt          -4.5761     0.6566  -6.969 9.16e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.494 on 20 degrees of freedom
Multiple R-squared:  0.7083, Adjusted R-squared:  0.6938
F-statistic: 48.57 on 1 and 20 DF, p-value: 9.159e-07
```

Figura 13.14: recta de mínimos cuadrados usando validación cruzada.

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

```
1 # Cargar los datos.
2 datos <- mtcars
```

```

3
4 # Crear conjuntos de entrenamiento y prueba.
5 set.seed(101)
6 n <- nrow(datos)
7 n_entrenamiento <- floor(0.7 * n)
8 muestra <- sample.int(n = n, size = n_entrenamiento, replace = FALSE)
9 entrenamiento <- datos[muestra, ]
10 prueba <- datos[-muestra, ]
11
12 # Ajustar modelo con el conjunto de entrenamiento.
13 modelo <- lm(mpg ~ wt, data = entrenamiento)
14 print(summary(modelo))
15
16 # Calcular error cuadrado promedio para el conjunto de entrenamiento.
17 mse_entrenamiento <- mean(modelo$residuals ** 2)
18 cat("MSE para el conjunto de entrenamiento:", mse_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[["mpg"]] - predicciones
25 mse_prueba <- mean(error ** 2)
26 cat("MSE para el conjunto de prueba:", mse_prueba)

```

Call:

```
lm(formula = mpg ~ wt, data = entrenamiento)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.7972	-2.7338	-0.0359	1.3380	6.5505

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	38.1048	2.3760	16.037	6.97e-13 ***
wt	-5.6044	0.7381	-7.593	2.59e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.075 on 20 degrees of freedom

Multiple R-squared: 0.7425, Adjusted R-squared: 0.7296

F-statistic: 57.66 on 1 and 20 DF, p-value: 2.586e-07

Figura 13.15: otra recta de mínimos cuadrados usando validación cruzada.

13.5.4 Validación cruzada de k pliegues

Una buena manera de mejorar la estimación del error cuadrático medio es obtener más observaciones, de acuerdo al ya conocido teorema del límite central. Para esto, se puede usar una nueva manera de remuestreo: la **validación cruzada de k pliegues** (en inglés *k-fold cross validation*). La idea de fondo es la misma de la validación cruzada expuesta en el apartado anterior: usar un conjunto de entrenamiento para ajustar el

modelo y otro de prueba para evaluarlo. Sin embargo, esta variante modifica este proceso a fin de obtener k estimaciones del error. Para ello se separa el conjunto de datos en k subconjuntos de igual tamaño y, como explica Amat Rodrigo (2016), realizamos k estimaciones del error cuadrático medio 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 $k - 1$ 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 errores cuadrados medios obtenidos en el paso 1.

En R, podemos realizar este proceso de forma bastante sencilla gracias a 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 `lm()`.
- **number**: cantidad de pliegues (k).

El lector atento habrá notado 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 modelos además de la RLS. El script 13.5 crea, una vez más, una RLS para predecir el rendimiento de un automóvil a partir de su peso, usando ahora validación cruzada de 5 pliegues.

Script 13.5: ajuste de una regresión lineal simple usando validación cruzada de 5 pliegues.

```

1 library(caret)
2
3 # Cargar los datos.
4 datos <- mtcars
5
6 # Crear conjuntos de entrenamiento y prueba.
7 set.seed(101)
8 n <- nrow(datos)
9 n_entrenamiento <- floor(0.7 * n)
10 muestra <- sample.int(n = n, size = n_entrenamiento, replace = FALSE)
11 entrenamiento <- datos[muestra, ]
12 prueba <- datos[-muestra, ]
13
14 # Ajustar modelo usando validación cruzada de 5 pliegues.
15 modelo <- train(mpg ~ wt, data = entrenamiento, method = "lm",
16                 trControl = trainControl(method = "cv", number = 5))
17
18 print(summary(modelo))
19
20 # Obtener error cuadrado promedio para el conjunto de entrenamiento.
21 mse_entrenamiento <- modelo$results$RMSE
22 cat("MSE para el conjunto de entrenamiento:", mse_entrenamiento, "\n")
23
24 # Hacer predicciones para el conjunto de prueba.
25 predicciones <- predict(modelo, prueba)
26
27 # Calcular error cuadrado promedio para el conjunto de prueba.
28 error <- prueba[["mpg"]] - predicciones
29 mse_prueba <- mean(error ** 2)
30 cat("MSE para el conjunto de prueba:", mse_prueba)

```

Un aspecto importante a tener en cuenta es que la función `train()` ajusta el modelo final con la totalidad del conjunto de entrenamiento, por lo que el error cuadrático medio para el conjunto de prueba y los parámetros del modelo son los mismos que ya habíamos obtenido. Sin embargo, la estimación del error cuadrático medio

para el conjunto de entrenamiento es diferente: al usar la semilla 101 se obtiene $MSE_e = 2,785$ y para la semilla 125, $MSE_e = 3,482$, valores bastante más parecidos entre sí.

13.5.5 Validación cruzada dejando uno fuera

Cuando la muestra disponible es pequeña, tema que reforzaremos en el capítulo siguiente, una buena alternativa es usar **validación cruzada dejando uno fuera** (*leave-one-out cross validation* en inglés). El esquema es el mismo que para validación cruzada con k pliegues, pero ahora usaremos tantos pliegues como observaciones tenga el conjunto de entrenamiento. En otras palabras, hacemos una iteración por cada elemento del conjunto de entrenamiento, reservando una única observación para validación. En R, la llamada a `train()` es muy similar a la que hicimos para validación cruzada con k pliegues: solo cambia el argumento `trControl`, cuyo valor ahora debe ser `trainControl(method = "LOOCV")`.

13.6 INFERENCIA PARA REGRESIÓN LINEAL

También podemos usar los modelos de RLS para hacer inferencia, procedimiento que ilustraremos mediante el siguiente ejemplo: el gerente de una empresa de desarrollo de software cree que, mientras más *stakeholders* tiene un proyecto, menos requisitos funcionales tiene el software a desarrollar. Para llevar a cabo el estudio pertinente, seleccionó aleatoriamente los datos de 15 proyectos de entre los 200 que ha desarrollado la empresa hasta la fecha, los cuales se muestran en la tabla 13.2.

requisitos	stakeholders
11	8
10	8
12	6
14	6
8	8
13	7
18	3
15	1
20	3
16	4
21	5
13	4
10	4
9	9
21	2

Tabla 13.2: requisitos funcionales y cantidad de *stakeholders* para diferentes proyectos desarrollados por la empresa.

Para llevar a cabo su estudio, el gerente ha ajustado un modelo de regresión lineal usando para ello el script 13.6. La recta ajustada y el gráfico de residuos se muestran en la figura 13.16.

Script 13.6: regresión lineal para la cantidad de requisitos funcionales de acuerdo a la cantidad de *stakeholders*.

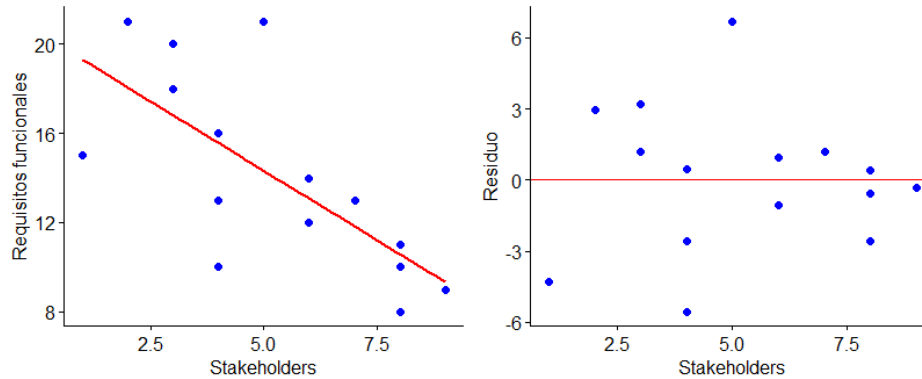


Figura 13.16: regresión lineal para la cantidad de requisitos funcionales de acuerdo a la cantidad de *stakeholders*.

```

1 library(ggpubr)
2
3 # Crear los datos originales.
4 requisitos <- c(11, 10, 12, 14, 8, 13, 18, 15, 20, 16, 21, 13, 10, 9, 21)
5 stakeholders <- c(8, 8, 6, 6, 8, 7, 3, 1, 3, 4, 5, 4, 4, 9, 2)
6 datos <- data.frame(requisitos, stakeholders)
7
8 # Ajustar modelo.
9 modelo <- lm(requisitos ~ stakeholders, data = datos)
10 print(summary(modelo))
11
12 # Graficar el modelo.
13 p <- ggscatter(
14   datos, x = "stakeholders", y = "requisitos", color = "blue", fill = "blue",
15   xlab = "Stakeholders", ylab = "Requisitos funcionales")
16
17 p <- p + geom_smooth(method = lm, se = FALSE, colour = "red")
18
19 # Graficar los residuos.
20 b_1 <- modelo$coefficients[2]
21 b_0 <- modelo$coefficients[1]
22 residuos <- datos[["requisitos"]] - (b_1 * datos[["stakeholders"]] + b_0)
23 datos <- data.frame(datos, residuos)
24
25 r <- ggscatter(datos, x = "stakeholders", y = "residuos", color = "blue",
26   fill = "blue", xlab = "Stakeholders", ylab = "Residuo")
27
28 r <- r + geom_hline(yintercept = 0, colour = "red")
29
30 g <- ggarrange(p, r, ncol = 2, nrow = 1)
31 print(g)
32
33 # Verificar normalidad de los residuos.
34 cat("Prueba de normalidad para los residuos\n")
35 print(shapiro.test(datos$residuos))

```

Puesto que la correlación entre ambas variables es relativamente fuerte ($R = -0,706$), podemos comprobar que los datos siguen una tendencia lineal. Al aplicar la prueba de normalidad de Shapiro-Wilk a los residuos, concluimos que estos siguen una distribución cercana a la normal ($p = 0,924$). Podemos apreciar en la figura 13.16 que la variabilidad de los residuos es relativamente constante. Por otra parte, las observaciones son

independientes entre sí, pues han sido seleccionadas de manera aleatoria y corresponden a menos del 10 % de la población. En consecuencia, se verifica el cumplimiento de todas las condiciones necesarias para emplear un modelo de RLS ajustado mediante mínimos cuadrados.

```
Call:
lm(formula = requisitos ~ stakeholders, data = datos)

Residuals:
    Min       1Q   Median       3Q      Max
-5.5624 -1.8160  0.4234  1.1840  6.6840

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   20.5482     1.9810  10.373 1.17e-07 ***
stakeholders  -1.2464     0.3466  -3.596 0.00326 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.184 on 13 degrees of freedom
Multiple R-squared:  0.4987, Adjusted R-squared:  0.4601
F-statistic: 12.93 on 1 and 13 DF,  p-value: 0.003255
```

Figura 13.17: descripción detallada del modelo obtenido por el gerente para el ejemplo.

En la descripción del modelo (figura 13.17) podemos notar, bajo el encabezado **Coefficients**, una tabla con dos filas: una por cada parámetro del modelo, donde la primera corresponde a la intercepción y la segunda, a la pendiente. A su vez, la primera columna identifica los parámetros del modelo y la segunda presenta sus valores estimados. Como toda estimación tiene asociado un margen de error, la tercera columna muestra el error estándar para cada parámetro. Las dos columnas restantes requieren de una explicación algo más detallada, por lo que no las describiremos aquí.

El gerente, con la intención de evaluar a una abatida estudiante en práctica que aún no ha cursado su asignatura de estadística, le ha entregado los resultados obtenidos y le ha preguntado si los datos sustentan su teoría de que la cantidad de requisitos funcionales disminuye a medida que la cantidad de *stakeholders* aumenta. Tras muchas horas buscando información, la estudiante ha formulado las siguientes hipótesis:

H_0 : $\beta_1 = 0$. La pendiente del modelo es igual a 0 o, lo que es lo mismo, la cantidad de *stakeholders* no explica en absoluto la cantidad de requisitos funcionales.

H_A : $\beta_1 < 0$.

Puesto que el valor p entregado por R corresponde a una prueba bilateral (fijarse en el valor absoluto que incluye el título de la columna: $\text{Pr}(>|t|)$), en el caso unilateral se debe considerar la mitad de este valor. En consecuencia, el gerente concluye, con 99 % de confianza ($p < 0.002$), que en efecto la cantidad de requisitos funcionales disminuye a medida que la cantidad de *stakeholders* aumenta.

13.7 EJERCICIOS PROPUESTOS

1. ¿Qué asume la regresión lineal, qué variables involucra y con qué parámetros trabaja?
2. ¿Cómo lucen los gráficos de dispersión de una relación lineal fuerte, de una débil y de una nula?
3. ¿Por qué hay que mirar un gráfico de dispersión de los datos al pensar en regresión lineal?
4. Describe cómo se usa la línea de regresión para predecir.

5. ¿Qué son los residuos? ¿Qué valores pueden tomar? ¿Qué utilidad tienen?
6. Explica qué mide la correlación, cómo se calcula y qué valores puede tomar.
7. Explica cómo funciona el método de los mínimos cuadrados.
8. ¿Qué condiciones necesita el método de los mínimos cuadrados para ser confiable?
9. ¿Cómo lucen los gráficos de residuos que no cumplen con alguno de los requisitos enunciados en el ejercicio anterior?
10. Explica cómo se interpretan los parámetros estimados con regresión por mínimos cuadrados.
11. Explica qué mide el coeficiente de determinación R^2 , cómo se calcula y qué valores puede tomar.
12. Explica cómo se interpretan los parámetros de la regresión lineal cuando la variable predictora es categórica.
13. Explica qué es apalancamiento y por qué es importante detectarlo.
14. ¿Cómo lucen los gráficos de datos (y residuos) que pueden tener problemas de apalancamiento?
15. ¿Cuáles son las hipótesis que se contrastan al hacer inferencia con la regresión lineal?
16. Investiga cómo se usa la función `lm()` de R y qué información entrega.

REFERENCIAS

- 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
- Diez, D., Barr, C. D. & Çetinkaya-Rundel, M. (2017). *OpenIntro Statistics* (3.^a ed.). <https://www.openintro.org/book/os/>.
- Field, A., Miles, J. & Field, Z. (2012). *Discovering statistics using R*. SAGE Publications Ltd.
- 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>