Regressão Linear

Fernando Náufel

03/05/2024 18:22

Índice

Apresentação

???

1 Regressão linear simples

1.1 Exemplo: vendas e publicidade

Exemplo baseado no livro James et al. (2021), com dados obtidos de https://www.kaggle.com/datasets/ashydv/advertising-dataset/data.

Este conjunto de dados contém 4 colunas:

- `tv`: verba (em milhares de dólares) gasta em publicidade na TV;
- `radio`: verba (em milhares de dólares) gasta em publicidade no rádio;
- `jornal`: verba (em milhares de dólares) gasta em publicidade em jornais;
- `vendas`: receita das vendas (em milhares de dólares).

Cada observação — isto é, cada linha — corresponde a um produto.

1.1.1 Leitura e limpeza

```
publicidade <- read_csv(
  'dados/advertising.csv',
  show_col_types = FALSE
) %>%
  janitor::clean_names() %>%
  rename(
    jornal = newspaper,
    vendas = sales
  )

publicidade %>% gt()
```

tv	radio	jornal	vendas
230,1	37,8	69,2	22,1
44,5	39,3	45,1	10,4
17,2	45,9	69,3	12,0

151,5	41,3	58,5	16,5
180,8	10,8	58,4	17,9
8,7	48,9	75,0	7,2
57,5	32,8	23,5	11,8
120,2	19,6	11,6	13,2
8,6	2,1	1,0	4,8
199,8	2,6	21,2	15,6
66,1	5,8	24,2	12,6
214,7	24,0	4,0	17,4
23,8	35,1	65,9	9,2
97,5	7,6	7,2	13,7
204,1	32,9	46,0	19,0
195,4	47,7	52,9	22,4
67,8	36,6	114,0	12,5
281,4	39,6	55,8	24,4
69,2	20,5	18,3	11,3
147,3	23,9	19,1	14,6
218,4	27,7	53,4	18,0
237,4	5,1	23,5	17,5
13,2	15,9	49,6	5,6
228,3	16,9	26,2	20,5
62,3	12,6	18,3	9,7
262,9	3,5	19,5	17,0
142,9	29,3	12,6	15,0
240,1	16,7	22,9	20,9
248,8	27,1	22,9	18,9
70,6	16,0	40,8	10,5
292,9	28,3	43,2	21,4
112,9	17,4	38,6	11,9
97,2	1,5	30,0	13,2
265,6	20,0	0,3	17,4
95,7	1,4	7,4	11,9
290,7	4,1	8,5	17,8
266,9	43,8	5,0	25,4
74,7	49,4	45,7	14,7
43,1	26,7	35,1	10,1
228,0	37,7	32,0	21,5
202,5	22,3	31,6	16,6
177,0	33,4	38,7	17,1
293,6	27,7	1,8	20,7
206,9	8,4	26,4	17,9
25,1	25,7	43,3	8,5
175,1	22,5	31,5	16,1

89,7	9,9	35,7	10,6
239,9	41,5	18,5	23,2
227,2	15,8	49,9	19,8
66,9	11,7	36,8	9,7
199,8	3,1	34,6	16,4
100,4	9,6	3,6	10,7
216,4	41,7	39,6	22,6
182,6	46,2	58,7	21,2
262,7	28,8	15,9	20,2
198,9	49,4	60,0	23,7
7,3	28,1	41,4	5,5
136,2	19,2	16,6	13,2
210,8	49,6	37,7	23,8
210,7	29,5	9,3	18,4
53,5	2,0	21,4	8,1
261,3	42,7	54,7	24,2
239,3	15,5	27,3	20,7
102,7	29,6	8,4	14,0
131,1	42,8	28,9	16,0
69,0	9,3	0,9	11,3
31,5	24,6	2,2	11,0
139,3	14,5	10,2	13,4
237,4	27,5	11,0	18,9
216,8	43,9	27,2	22,3
199,1	30,6	38,7	18,3
109,8	14,3	31,7	12,4
26,8	33,0	19,3	8,8
129,4	5,7	31,3	11,0
213,4	24,6	13,1	17,0
16,9	43,7	89,4	8,7
27,5	1,6	20,7	6,9
120,5	28,5	14,2	14,2
5,4	29,9	9,4	5,3
116,0	7,7	23,1	11,0
76,4	26,7	22,3	11,8
239,8	4,1	36,9	17,3
75,3	20,3	32,5	11,3
68,4	44,5	35,6	13,6
213,5	43,0	33,8	21,7
193,2	18,4	65,7	20,2
76,3	27,5	16,0	12,0
110,7	40,6	63,2	16,0
88,3	25,5	73,4	12,9

109,8	47,8	51,4	16,7
134,3	4,9	9,3	14,0
28,6	1,5	33,0	7,3
217,7	33,5	59,0	19,4
250,9	36,5	72,3	22,2
107,4	14,0	10,9	11,5
163,3	31,6	52,9	16,9
197,6	3,5	5,9	16,7
184,9	21,0	22,0	20,5
289,7	42,3	51,2	25,4
135,2	41,7	45,9	17,2
222,4	4,3	49,8	16,7
296,4	36,3	100,9	23,8
280,2	10,1	21,4	19,8
187,9	17,2	17,9	19,7
238,2	34,3	5,3	20,7
137,9	46,4	59,0	15,0
25,0	11,0	29,7	7,2
90,4	0,3	23,2	12,0
13,1	0,4	25,6	5,3
255,4	26,9	5,5	19,8
225,8	8,2	56,5	18,4
241,7	38,0	23,2	21,8
175,7	15,4	2,4	17,1
209,6	20,6	10,7	20,9
78,2	46,8	34,5	14,6
75,1	35,0	52,7	12,6
139,2	14,3	25,6	12,2
76,4	0,8	14,8	9,4
125,7	36,9	79,2	15,9
19,4	16,0	22,3	6,6
141,3	26,8	46,2	15,5
18,8	21,7	50,4	7,0
224,0	2,4	15,6	16,6
123,1	34,6	12,4	15,2
229,5	32,3	74,2	19,7
87,2	11,8	25,9	10,6
7,8	38,9	50,6	6,6
80,2	0,0	9,2	11,9
220,3	49,0	3,2	24,7
59,6	12,0	43,1	9,7
0,7	39,6	8,7	1,6
265,2	2,9	43,0	17,7

8,4	27,2	2,1	5,7
219,8	33,5	45,1	19,6
36,9	38,6	65,6	10,8
48,3	47,0	8,5	11,6
25,6	39,0	9,3	9,5
273,7	28,9	59,7	20,8
43,0	25,9	20,5	9,6
184,9	43,9	1,7	20,7
73,4	17,0	12,9	10,9
193,7	35,4	75,6	19,2
220,5	33,2	37,9	20,1
104,6	5,7	34,4	10,4
96,2	14,8	38,9	12,3
140,3	1,9	9,0	10,3
240,1	7,3	8,7	18,2
243,2	49,0	44,3	25,4
38,0	40,3	11,9	10,9
44,7	25,8	20,6	10,1
280,7	13,9	37,0	16,1
121,0	8,4	48,7	11,6
197,6	23,3	14,2	16,6
171,3	39,7	37,7	16,0
187,8	21,1	9,5	20,6
4,1	11,6	5,7	3,2
93,9	43,5	50,5	15,3
149,8	1,3	24,3	10,1
11,7	36,9	45,2	7,3
131,7	18,4	34,6	12,9
172,5	18,1	30,7	16,4
85,7	35,8	49,3	13,3
188,4	18,1	25,6	19,9
163,5	36,8	7,4	18,0
117,2	14,7	5,4	11,9
234,5	3,4	84,8	16,9
17,9	37,6	21,6	8,0
206,8	5,2	19,4	17,2
215,4	23,6	57,6	17,1
284,3	10,6	6,4	20,0
50,0	11,6	18,4	8,4
164,5	20,9	47,4	17,5
19,6	20,1	17,0	7,6
168,4	7,1	12,8	16,7
222,4	3,4	13,1	16,5

276,9	48,9	41,8	27,0
248,4	30,2	20,3	20,2
170,2	7,8	35,2	16,7
276,7	2,3	23,7	16,8
165,6	10,0	17,6	17,6
156,6	2,6	8,3	15,5
218,5	5,4	27,4	17,2
56,2	5,7	29,7	8,7
287,6	43,0	71,8	26,2
253,8	21,3	30,0	17,6
205,0	45,1	19,6	22,6
139,5	2,1	26,6	10,3
191,1	28,7	18,2	17,3
286,0	13,9	3,7	20,9
18,7	12,1	23,4	6,7
39,5	41,1	5,8	10,8
75,5	10,8	6,0	11,9
17,2	4,1	31,6	5,9
166,8	42,0	3,6	19,6
149,7	35,6	6,0	17,3
38,2	3,7	13,8	7,6
94,2	4,9	8,1	14,0
177,0	9,3	6,4	14,8
283,6	42,0	66,2	25,5
232,1	8,6	8,7	18,4

1.1.2 Divisão em dados de treino e teste

```
split <- initial_split(publicidade)
treino <- training(split)
teste <- testing(split)
split
```

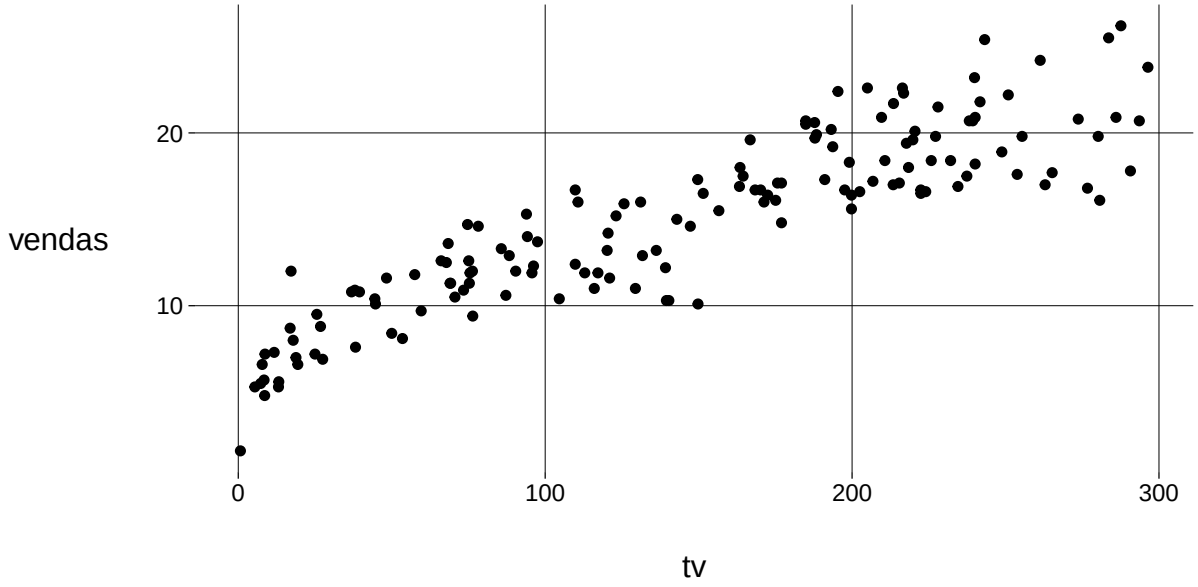
```
<Training/Testing/Total>
<150/50/200>
```

1.1.3 Vendas por verba gasta em TV

1.1.3.1 Análise exploratória

Começamos visualizando os dados:

```
grafico <- treino %>%
  ggplot(aes(tv, vendas)) +
    geom_point()

grafico
```



A correlação linear entre vendas e tv é

```
cor(treino$vendas, treino$tv)
```

```
[1] 0,8919795
```

1.1.3.2 Modelo linear

```r
modelo <- lm(vendas ~ tv, data = treino)
summary(modelo)
```

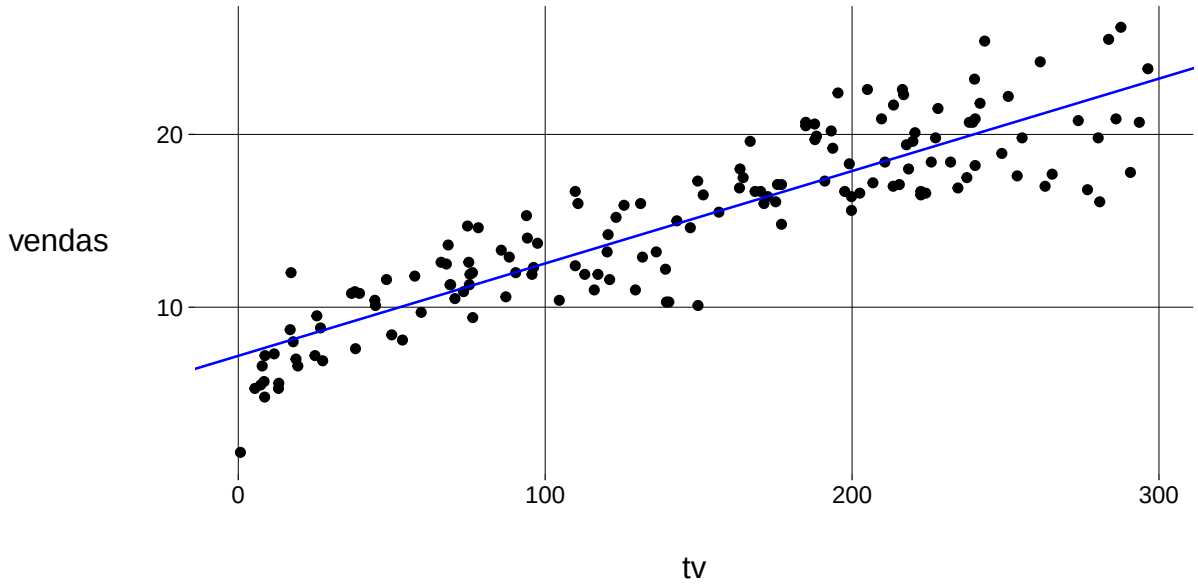
```
Call:
lm(formula = vendas ~ tv, data = treino)

Residuals:
    Min      1Q  Median      3Q     Max
-6,0968 -1,5960 -0,0152  1,6301  5,2086

Coefficients:
            Estimate Std. Error t value            Pr(>|t|)
(Intercept) 7,185427   0,373754   19,23 <0,0000000000000002 ***
tv          0,053478   0,002228   24,00 <0,0000000000000002 ***
---
Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1

Residual standard error: 2,289 on 148 degrees of freedom
Multiple R-squared:  0,7956,    Adjusted R-squared:  0,7942
F-statistic: 576,2 on 1 and 148 DF,  p-value: < 0,00000000000000022
```

```r
modelo_tidy <- tidy(modelo)
modelo_tidy
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   7.19     0.374        19.2 4.49e-42
2 tv            0.0535   0.00223      24.0 6.86e-53
```

```r
b0 <- modelo_tidy$estimate[1]
b1 <- modelo_tidy$estimate[2]
```

```r
grafico +
  geom_abline(
    intercept = b0,
    slope = b1,
    color = 'blue'
  )
```



A equação da reta é

$$\begin{aligned}\widehat{\text{vendas}} &= \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{tv} \\ &= 7{,}19 + 0{,}05 \cdot \text{tv}\end{aligned}$$

1.2 Teoria

1.2.1 Estimativas $\hat{\beta}_0$ e $\hat{\beta}_1$

Os valores achados são estimativas para β_0 e β_1, baseadas nos dados do conjunto de treino.

Por isso, os valores de `vendas` obtidos com esta equação também são estimativas.

Vamos escrever estimativas com o acento circunflexo (chapéu) sobre os símbolos.

De onde vêm os valores de $\hat{\beta}_0$ e $\hat{\beta}_1$?

Resposta: são os valores que fazem com que a soma dos quadrados das distâncias verticais dos pontos à reta seja a menor possível.

(Estas distâncias são chamadas de resíduos.)

Consulte este material para ver os detalhes sobre o cálculo de $\hat{\beta}_0$ e $\hat{\beta}_1$.

1.2.2 Erros-padrão das estimativas

Vamos pensar nas incertezas associadas aos valores de $\hat{\beta}_0$ e $\hat{\beta}_1$, com base na excelente discussão em (De Veaux, Velleman e Bock 2016, cap. 25).

Quais são os fatores que afetam a nossa confiança na reta de regressão?

Mais especificamente, quais os fatores que afetam nossa confiança no valor estimado $\hat{\beta}_1$ (a inclinação da reta)?

1.2.2.1 Espalhamento dos pontos em volta da reta

Quanto mais afastados da reta estiverem os dados, menor a nossa confiança de que a reta captura a variação de uma variável em função da outra.

Observe a Figura 1.1. O gráfico da esquerda nos dá mais certeza de que uma reta de regressão terá uma inclinação bem próxima da taxa de variação de y em função de x na população.

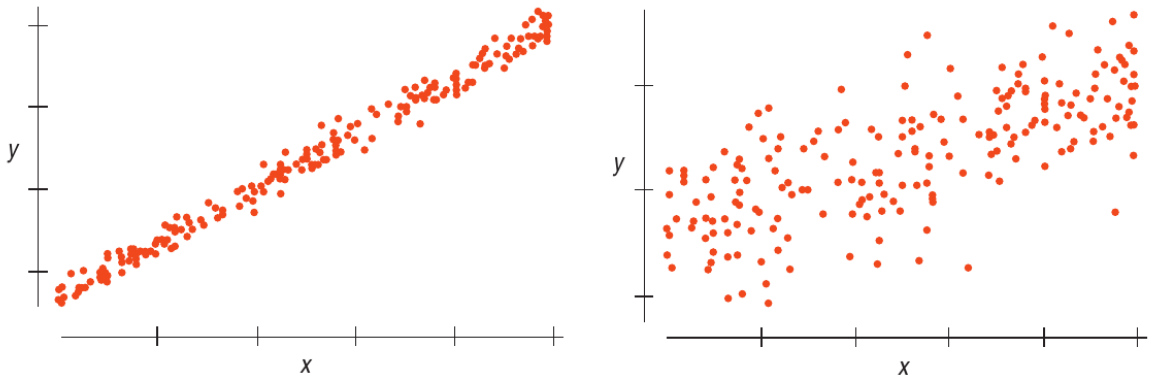


Figura 1.1: Espalhamento dos pontos

Este espalhamento é medido pelo desvio-padrão dos resíduos.

No exemplo das vendas, este desvio-padrão dos resíduos é calculado como

$$\sqrt{\frac{\sum_i (\text{vendas}_i - \widehat{\text{vendas}}_i)^2}{n-2}}$$

No numerador, o valor $\text{vendas}_i - \widehat{\text{vendas}}_i$ é o resíduo da observação i.

As `vendas` estimadas para cada valor de `tv` e os valores dos resíduos podem ser acessados assim:

```r
modelo_augment <- augment(modelo)
modelo_augment %>%
  select(vendas, tv, .fitted, .resid)
```

```
# A tibble: 150 x 4
  vendas    tv .fitted .resid
   <dbl> <dbl>   <dbl>  <dbl>
1   20.9 240.    20.0   0.874
2   11   116     13.4  -2.39
3   10.1  44.7    9.58  0.524
4   16.7 222.    19.1  -2.38
5   18.4 226.    19.3  -0.861
6    9.5  25.6    8.55  0.946
# i 144 more rows
```

Calculando o desvio-padrão dos resíduos:

```r
n <- nrow(modelo_augment)
dp_residuos <- sqrt(sum(modelo_augment$.resid^2) / (n - 2))
dp_residuos
```

```
[1] 2,28897
```

Este valor pode ser obtido na coluna `sigma` do *data frame* retornado pela função `glance`:

```r
modelo_glance <- glance(modelo)
modelo_glance$sigma
```

```
[1] 2,28897
```

! Desvio-padrão dos resíduos

No geral, então, em uma regressão da variável y sobre a variável x com n observações, o desvio-padrão dos resíduos é

$$s_{\text{residuos}} = \sqrt{\frac{\sum_i (y_i - \hat{y}_i)^2}{n-2}}$$

Pela Figura 1.1 e pelos comentários acima, quanto maior o valor de s_{residuos}, maior a

nossa incerteza.

1.2.2.2 Espalhamento de x

Quanto maior o espalhamento dos valores de x, maior nossa confiança na reta de regressão, pois ela estará baseada em uma diversidade maior de valores.

Observe a Figura 1.2. O gráfico da direita tem um espalhamento maior dos valores de x. Uma reta de regressão, ali, parece estar mais bem "ancorada".

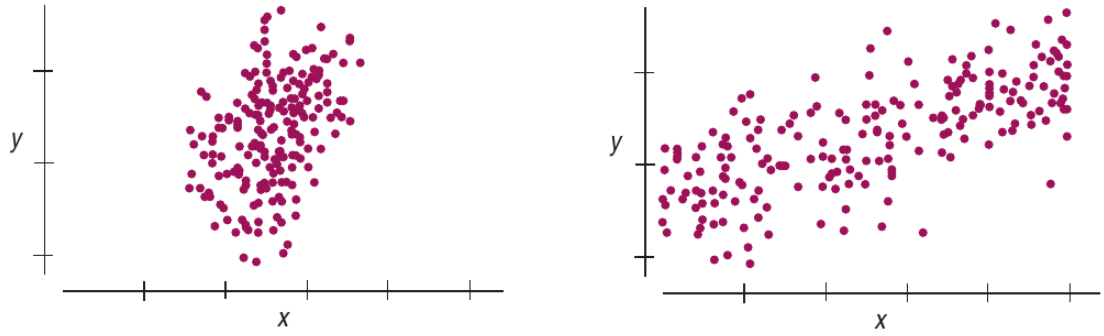


Figura 1.2: Espalhamento de x

O espalhamento de x é medido pelo desvio-padrão, que é calculado da maneira usual.

No exemplo das vendas, s_x, o desvio-padrão de `tv` é

```
dp_x <- modelo_augment %>%
  pull(tv) %>%
  sd()

dp_x
```

```
[1] 84,16717
```

! Desvio-padrão dos resíduos

Pela Figura 1.2 e pelos comentários acima, quanto maior o valor de s_x, menor a nossa incerteza.

1.2.2.3 Quantidade de dados

Uma reta baseada em mais pontos é mais confiável. Observe a Figura 1.3.

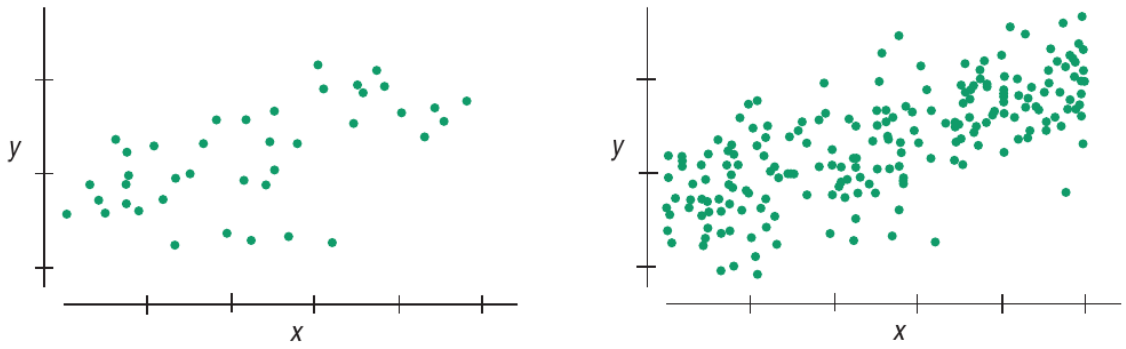


Figura 1.3: Quantidade de dados

> **! Quantidade de dados**
>
> Pela Figura 1.3 e pelos comentários acima, quanto maior o valor de n, menor a nossa incerteza.

1.2.2.4 Juntando tudo

Vimos que

- Quanto maior o desvio-padrão dos resíduos (s_{residuos}), maior a incerteza.
- Quanto maior o desvio-padrão da variável x (s_x), menor a incerteza.
- Quanto maior a quantidade de dados (n), menor a incerteza.

Concluímos que a incerteza sobre nossa estimativa para β_1 (a inclinação da reta) é proporcional aos valores acima da seguinte maneira:

$$EP(\beta_1) \propto \frac{s_{\text{residuos}}}{n \cdot s_x}$$

onde estamos escrevendo a incerteza como $EP(\beta_1)$, o erro-padrão de β_1.

! Erro-padrão de β_1

A fórmula exata para a incerteza sobre β_1 é

$$EP(\beta_1) = \frac{s_{\text{residuos}}}{\sqrt{n-1} \cdot s_x}$$

No exemplo das vendas, usando as variáveis que já calculamos antes, este erro-padrão é

```
dp_residuos / (sqrt(n - 1) * dp_x)
```

```
[1] 0,002227943
```

Este valor aparece nos resultados de `lm` como `std.error`:

```
modelo_tidy
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   7.19     0.374        19.2 4.49e-42
2 tv            0.0535   0.00223      24.0 6.86e-53
```

1.2.2.5 Erro-padrão do intercepto

! Erro-padrão de β_0

Para o intercepto β_0, o raciocínio é análogo.
A fórmula exata para a incerteza sobre β_0 é

$$EP(\beta_0) =$$

??? ISLR p. 76

1.3 Visão geométrica

Faraway (2016)

1.3.1 Um pequeno exemplo

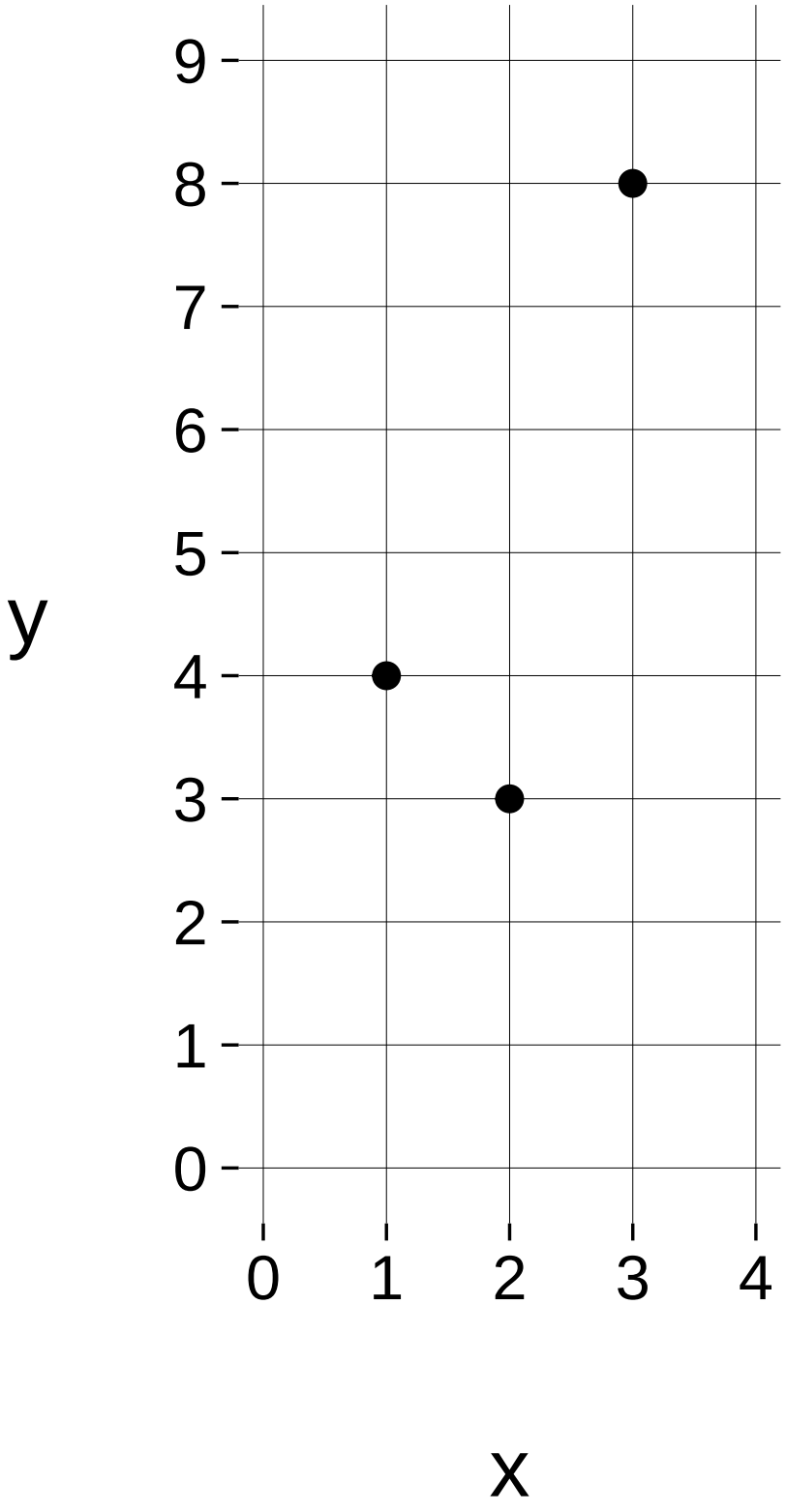
Para podermos visualizar a geometria, vamos considerar um conjunto de dados com apenas 3 observações.

A variável `x` é o único preditor, e a variável `y` é a resposta.

```
df <- tibble(
  x = 1:3,
  y = c(4, 3, 8)
)
```

x	y
1	4
2	3
3	8

Graficamente:


y
9
8
7
6
5
4
3
2
1
0
0
1
2
3
4
x

Com um único preditor, este é um exemplo de regressão simples. Queremos achar uma equação da forma

$$\hat{y} = \beta_0 + \beta_1 x$$

com valores de β_0 e β_1 que garantam a menor soma dos quadrados dos resíduos.

Usamos o R para achar os coeficientes e outras informações sobre este modelo:

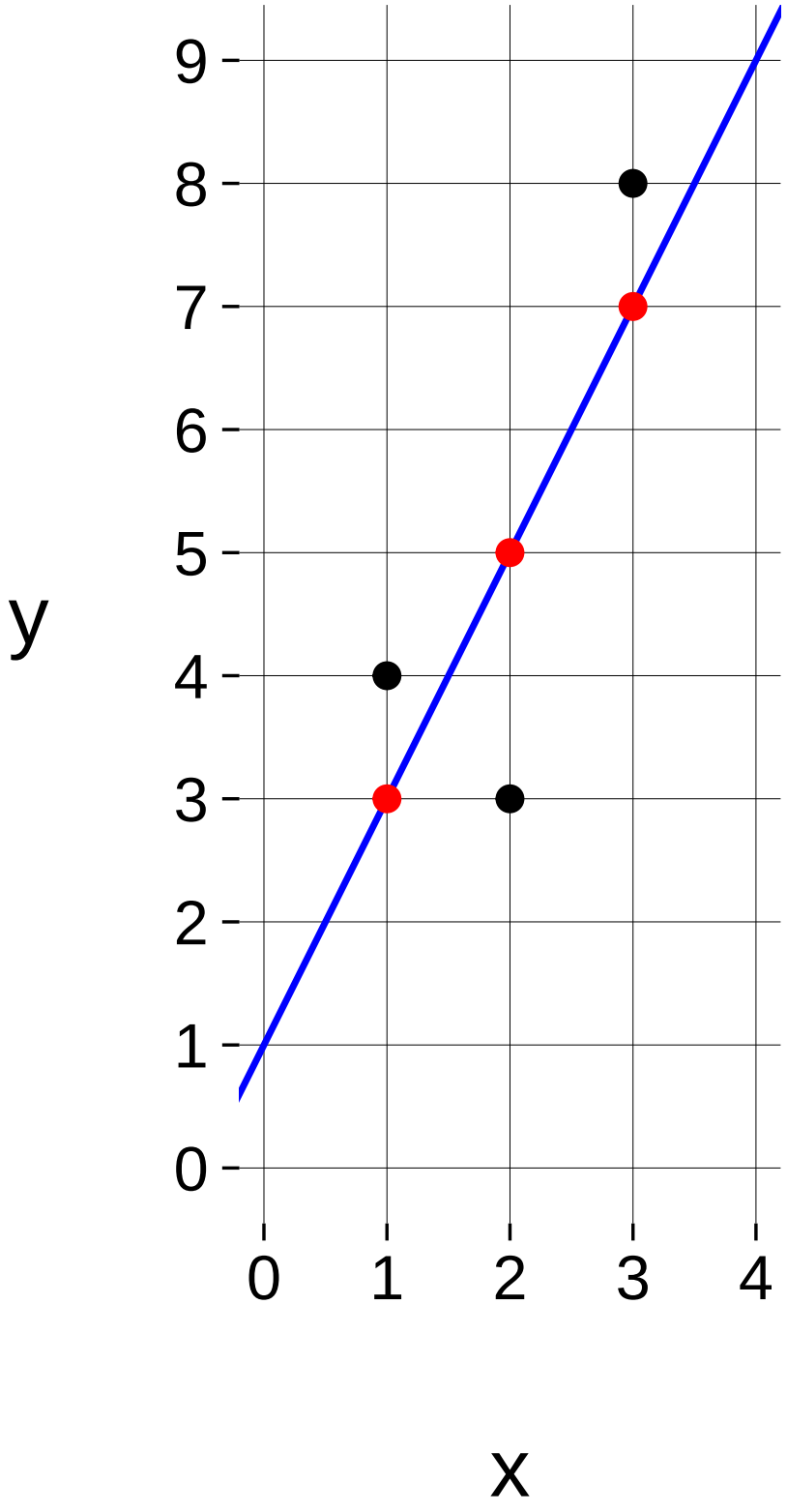
```
modelo <- lm(y ~ x, df)
summary(modelo)
```

```
Call:
lm(formula = y ~ x, data = df)

Residuals:
 1  2  3
 1 -2  1

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    1,000      3,742   0,267    0,834
x              2,000      1,732   1,155    0,454

Residual standard error: 2,449 on 1 degrees of freedom
Multiple R-squared:  0,5714,    Adjusted R-squared:  0,1429
F-statistic: 1,333 on 1 and 1 DF,  p-value: 0,4544
```

A equação da reta que procuramos é

$$\hat{y} = 1{,}00 + 2{,}00x$$

No gráfico, os valores de $\hat{y}$, para cada valor de x, são mostrados em vermelho. A reta de regressão é mostrada em azul:


y
9
8
7
6
5
4
3
2
1
0
0
1
2
3
4
x

Os valores de y, os valores previstos e os resíduos são

x	y	previsto	resíduo
1	4	3	1
2	3	5	-2
3	8	7	1

Usando Álgebra Linear, vamos encarar este modelo de outra forma.

A coluna `y` dos dados é representada pelo vetor

$$\mathbf{Y} = \begin{bmatrix} 4 \\ 3 \\ 8 \end{bmatrix}$$

Vamos definir a seguinte matriz:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$$

Nesta matriz, a segunda coluna corresponde à coluna `x` dos dados. A primeira coluna, com valores 1, está ali para podermos escrever o modelo como a equação matricial

$$\hat{\mathbf{Y}} = \mathbf{X} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

que, de forma mais detalhada, é

$$\begin{bmatrix} \widehat{y_1} \\ \widehat{y_2} \\ \widehat{y_3} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

ou, ainda,

$$\begin{bmatrix} \widehat{y_1} \\ \widehat{y_2} \\ \widehat{y_3} \end{bmatrix} = \begin{bmatrix} \beta_0 + & \beta_1 \\ \beta_0 + 2 \cdot & \beta_1 \\ \beta_0 + 3 \cdot & \beta_1 \end{bmatrix}$$

ou, explicitando os vetores que correspondem às colunas da matriz $\mathbf{X}$:

$$\begin{bmatrix} \widehat{y_1} \\ \widehat{y_2} \\ \widehat{y_3} \end{bmatrix} = \beta_0 \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \beta_1 \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \tag{1.1}$$

Agora, as considerações geométricas:

1. As colunas `x` e `y` do conjunto de dados são vetores com 3 componentes, que vivem em $\mathbb{R}^3$.
2. O vetor $\hat{\mathbf{Y}}$ também tem 3 componentes, mas a Equação 1.1 está dizendo que $\hat{\mathbf{Y}}$ é uma combinação linear dos dois vetores (linearmente independentes) $[1\ 1\ 1]^T$ e $[1\ 2\ 3]^T$.
3. Os dois vetores $[1\ 1\ 1]^T$ e $[1\ 2\ 3]^T$ não são capazes de gerar todo o espaço $\mathbb{R}^3$; o espaço gerado por eles é um plano.
4. O vetor $\mathbf{Y}$ (com os valores verdadeiros da variável de resposta y) não está no plano gerado pelos vetores $[1\ 1\ 1]^T$ e $[1\ 2\ 3]^T$ (verifique).
5. A relação verdadeira entre $\mathbf{Y}$ e $\mathbf{X}$ é

   $$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \beta_0 \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \beta_1 \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}$$

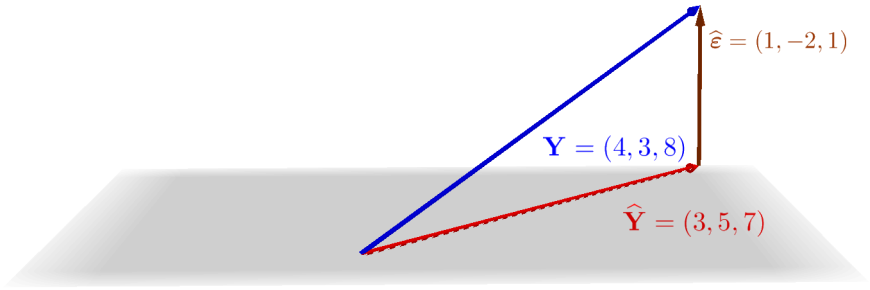
   onde os valores ε_i são os erros que o modelo não consegue capturar.
6. Estes erros ε_i são estimados pelos resíduos $\widehat{\varepsilon_i}$, de maneira que podemos escrever

   $$\begin{bmatrix} \widehat{y_1} \\ \widehat{y_2} \\ \widehat{y_3} \end{bmatrix} = \beta_0 \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \beta_1 \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} \widehat{\varepsilon_1} \\ \widehat{\varepsilon_2} \\ \widehat{\varepsilon_3} \end{bmatrix}$$

   O vetor de resíduos é

   $$\hat{} = \begin{bmatrix} \widehat{\varepsilon_1} \\ \widehat{\varepsilon_2} \\ \widehat{\varepsilon_3} \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

A situação é mostrada na figura:



$$(x, y, z) = r \cdot (1, 1, 1) + s \cdot (1, 2, 3)$$

O plano cinza é o espaço gerado pelos vetores $[1\ 1\ 1]^T$ e $[1\ 2\ 3]^T$. Na equação paramétrica deste plano, r e s correspondem aos valores possíveis de β_0 e β_1, respectivamente.

O vetor $\hat{\mathbf{Y}}$ (dos valores previstos pelo modelo) é a projeção ortogonal do vetor $\mathbf{Y}$ (dos valores verdadeiros da variável de resposta) sobre o plano gerado pelas colunas da matriz $\mathbf{X}$. Mais abaixo, vamos ver os detalhes desta projeção. O importante é entender que, quaisquer que sejam os valores de β_0 e β_1, o vetor $\hat{\mathbf{Y}}$ de valores previstos vai estar sempre limitado ao plano gerado pelas colunas da matriz $\mathbf{X}$.

Isto corresponde à intuição de que estamos perdendo informação ao tentar representar objetos de dimensão 3 (o número de observações do conjunto de dados) em um espaço de dimensão 2 (o número de parâmetros do modelo: β_0 e β_1).

???

2 Regressão linear múltipla

2.1 Simulação

2.1.1 Multicolinearidade

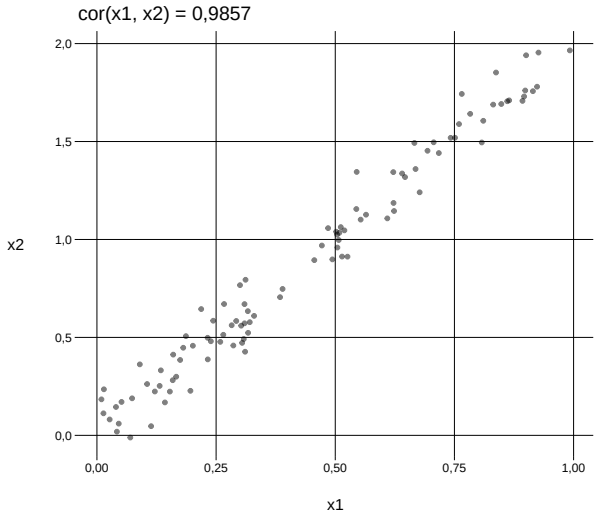
Vamos criar três preditores `x1`, `x2` e `x3`, com os dois primeiros correlacionados:

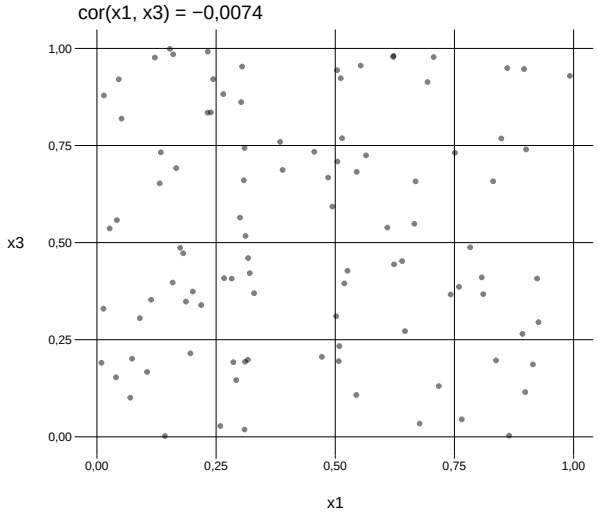
```
n <- 100
a <- 2
x1 <- runif(n)
x2 <- a * x1 + rnorm(n, 0, .1)
x3 <- runif(n)

df <- tibble(x1, x2, x3)
```

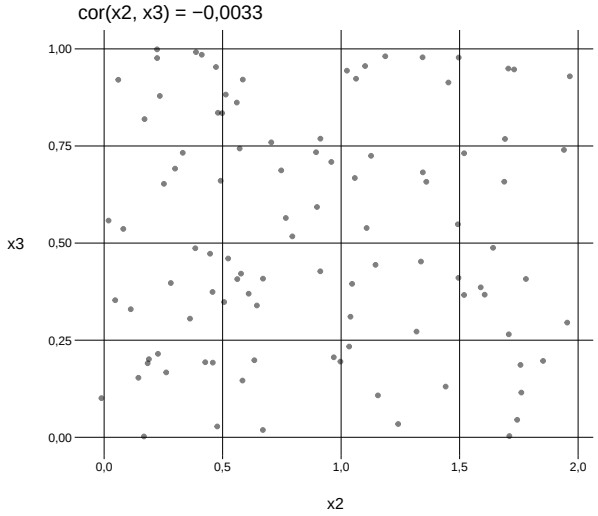
Gráficos:

```
plot_cor <- function(df, v1, v2) {

  x = df[[v1]]
  y = df[[v2]]
  valor_cor <- cor(x, y) %>% round(4)

  df %>% ggplot(aes(x, y)) +
    geom_point(alpha = 0.5) +
    labs(
      title = paste0('cor(', v1, ', ', v2, ') = ', valor_cor),
      x = v1,
      y = v2
    )

}
```

```r
v <- c('x1', 'x2', 'x3')

pares <- expand_grid(x = v, y = v) %>%
  filter(x < y) %>%
  arrange(x, y)

v1 <- pares %>% pull(x)
v2 <- pares %>% pull(y)

plots <- map2(
  v1, v2, ~ plot_cor(df, .x, .y)
)

plots %>%
  wrap_plots(
    ncol = 1,
    byrow = TRUE
  )
```


cor(x1, x2) = 0,9857
2,0
1,5
1,0
0,5
0,0
x2
0,00
0,25
0,50
0,75
1,00
x1


cor(x1, x3) = −0,0074
1,00
0,75
0,50
0,25
0,00
x3
0,00
0,25
0,50
0,75
1,00
x1


cor(x2, x3) = −0,0033
1,00
0,75
0,50
0,25
0,00
x3
0,0
0,5
1,0
1,5
2,0
x2

A variável de resposta é y:

```
b0 <- 1
b1 <- 2
b2 <- 3
b3 <- 4
var_epsilon <- .5

y <- b0 + b1 * x1 + b2 * x2 + b3 * x3 + rnorm(n, sd = sqrt(var_epsilon))
df_y <- df %>%
  mutate(y = y)
```

Usando todas as variáveis, temos:

```
modelo_123 <- lm(y ~ ., data = df_y)
```

A equação verdadeira é

$$y = 1 + 2x_1 + 3x_2 + 4x_3 + \varepsilon$$

O modelo deu os coeficientes

```
modelo_123 %>% summary()
```

```
Call:
lm(formula = y ~ ., data = df_y)

Residuals:
     Min       1Q   Median       3Q      Max
-2,39333 -0,42820  0,08625  0,38997  1,83332

Coefficients:
            Estimate Std. Error t value             Pr(>|t|)
(Intercept)   1,2345     0,1747   7,068       0,000000000252 ***
x1            3,2787     1,4582   2,249              0,02683 *
x2            2,3932     0,7203   3,323              0,00126 **
x3            3,7137     0,2286  16,245 < 0,0000000000000002 ***
---
Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
```

```
Residual standard error: 0,681 on 96 degrees of freedom
Multiple R-squared:  0,9334,    Adjusted R-squared:  0,9313
F-statistic: 448,6 on 3 and 96 DF,  p-value: < 0,00000000000000022
```

```
modelo_123
```

```

Call:
lm(formula = y ~ ., data = df_y)

Coefficients:
(Intercept)           x1           x2           x3
      1,235        3,279        2,393        3,714
```

Agora, usando apenas `x1` e `x3`:

```
modelo_13 <- lm(y ~ x1 + x3, data = df_y)
```

A equação verdadeira é — substituindo x_2 por $(b_1 + ab_2)x_1$ —

$$y = 1 + 8x_1 + 4x_3 + \varepsilon$$

O modelo deu os coeficientes

```
modelo_13 %>% summary()
```

```

Call:
lm(formula = y ~ x1 + x3, data = df_y)

Residuals:
    Min      1Q  Median      3Q     Max
-2,2807 -0,4648  0,0890  0,4492  1,8246

Coefficients:
            Estimate Std. Error t value             Pr(>|t|)
(Intercept)   1,2489     0,1834   6,808       0,000000000825 ***
x1            8,0547     0,2580  31,224 < 0,0000000000000002 ***
x3            3,7315     0,2401  15,543 < 0,0000000000000002 ***
---
```

```
Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1

Residual standard error: 0,7153 on 97 degrees of freedom
Multiple R-squared:  0,9258,    Adjusted R-squared:  0,9242
F-statistic: 604,7 on 2 and 97 DF,  p-value: < 0,00000000000000022
```

```
modelo_13
```

```
Call:
lm(formula = y ~ x1 + x3, data = df_y)

Coefficients:
(Intercept)           x1           x3
      1,249        8,055        3,732
```

Em termos do R^2 ajustado:

- O modelo com os três preditores teve $R^2_{\text{adj}} = 0{,}9313$.
- O modelo com dois preditores teve $R^2_{\text{adj}} = 0{,}9242$.

Para a equação verdadeira:

```
y_eq <- b0 + b1 * x1 + b2 * x2 + b3 * x3
rsq_vec(y, y_eq)
```

```
[1] 0,931878
```

Anova diz que o segundo modelo é mais significativo que o primeiro:

```
anova(modelo_123, modelo_13)
```

```
# A tibble: 2 x 6
  Res.Df   RSS    Df `Sum of Sq`     F `Pr(>F)`
   <dbl> <dbl> <dbl>       <dbl> <dbl>    <dbl>
1     96  44.5    NA        NA     NA   NA
2     97  49.6    -1        -5.12  11.0  0.00126
```

```r
modelo_123 %>% glance()
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl>
1     0.933         0.931 0.681      449. 2.54e-56     3  -101.  213.
# i 4 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>,
#   nobs <int>
```

```r
modelo_13 %>% glance()
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl>
1     0.926         0.924 0.715      605. 1.69e-55     2  -107.  222.
# i 4 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>,
#   nobs <int>
```

Referências

De Veaux, R. D., P. F. Velleman, e D. E. Bock. 2016. *Stats: Data and Models*. 4.ª ed. Pearson Education. https://media.pearsoncmg.com/aw/aw_deveaux_stats_4_2016/websites/statdm4d_comp_web_launch.html.

Faraway, Julian J. 2016. *Linear Models with R*. 2.ª ed. Chapman; Hall/CRC. https://doi.org/10.1201/b17144.

James, Gareth, Daniela Witten, Trevor Hastie, e Robert Tibshirani. 2021. *An Introduction to Statistical Learning: With Applications in R*. 2.ª ed. Springer Publishing Company, Incorporated. https://www.statlearning.com/.