

# Regresión lineal simple

Guillermo Ayala Gallego

## Regresión lineal simple

Guillermo Ayala Gallego

2024-03-25

## Recta de mínimos cuadrados

\(\dots\)

- Nuestros datos son  $((x_i, y_i))$  con  $(i=1, \dots, n)$ .
- Un modelo sencillo podría ser

$$[ y_i = \beta_0 + \beta_1 x_i, \quad i = 1, \dots, n ]$$

- No es posible.
- Una **buena** solución aproximada.
- Nos conformamos con

$$[ y_i \approx \beta_0 + \beta_1 x_i, \quad i = 1, \dots, n ]$$

\(\dots\)

- Una posibilidad para determinar unos **buenos** valores para  $(\beta_0)$  y  $(\beta_1)$  es considerar y encontrar los valores de los coeficientes que hacen mínima esta expresión.

$$[ S_t = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2. ]$$

- Haciendo algún cálculo se obtiene

$$[ \hat{\beta}_1 = \frac{\sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)/n}{\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2/n}. ]$$

$$[ \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}. ]$$

## Recta de mínimos cuadrados\(\dots\)

- Si denotamos

$$S_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}),$$

y

$$S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2, \quad S_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2,$$

entonces

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}.$$

## Predicción y residuo

- Nos planteamos encontrar unas buenas aproximaciones para  $(y_i)$  utilizando  $(x_i)$ .
- Lo natural sería tomar el siguiente valor la **predicción** o **valor ajustado** para  $(y_i)$

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i. \quad \text{- El residuo es}$$

$$e_i = y_i - \hat{y}_i.$$

## Coefficiente de determinación: $(R^2)$

- Es una medida de la calidad del ajuste.

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = \frac{SS(\text{regression})}{SS(\text{total})}.$$

- Se tiene

$$R^2 = \frac{S_{xy}^2}{S_{xx}S_{yy}},$$

que no es más que el cuadrado del coeficiente de correlación entre la predictora y la respuesta.

## Modelo

### Regresión lineal simple

- $(Y_i = \beta_0 + \beta_1 x_i + \epsilon_i)$  para  $(i=1, \dots, n)$ .
- $(\epsilon_i \sim N(0, \sigma^2))$ .
- Los errores  $(\epsilon_i)$  son independientes.

### Regresión lineal simple $(\dots)$

- Estamos asumiendo que la distribución **condicionada** de  $(Y_i)$  al predictor  $(x_i)$  es normal con media  $(\beta_0 + \beta_1 x_i)$  y varianza  $(\sigma^2)$ ,

$\{Y_i | x_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2), \}$

y que los distintos  $\{Y_i\}$  son independientes entre sí.

## Representación matricial

- Tenemos el **vector de respuestas aleatorias**,  $\{\mathbf{Y}\}$ ; la **matriz de modelo**,  $\{\mathbf{X}\}$ ; el **vector de coeficientes**,  $\{\mathbf{\beta}\}$  y el **vector de errores aleatorios**,  $\{\mathbf{\epsilon}\}$  dados por

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}; \mathbf{X} = \begin{bmatrix} 1 & x_1 & \dots & 1 & x_n \end{bmatrix}; \mathbf{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}; \mathbf{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

- El modelo de regresión lineal simple se puede formular como  $\mathbf{Y} = \mathbf{X} \mathbf{\beta} + \mathbf{\epsilon}$ .

- El vector de errores aleatorios  $\{\mathbf{\epsilon}\}$  sigue una distribución normal multivariante  $\{\mathbf{\epsilon} \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)\}$ .

## Verosimilitud

- La función de verosimilitud viene dada por 
$$L(\beta_0, \beta_1, \sigma^2; y_1, \dots, y_n) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{1}{2\sigma^2} (y_i - \beta_0 - \beta_1 x_i)^2} = \frac{1}{(2\pi)^{n/2} \sigma^n} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2}$$
- Los estimadores máximo verosímiles de los coeficientes corresponden con los estimadores mínimo cuadráticos.

## Paquetes

- Cargamos los paquetes que necesitamos.

```
1 pacman::p_load(Biobase, ggplot2)
```

## Datos

- Leemos los datos.
- La primera es la más ortodoxa.

```
1 data(gse25171, package="tamidata2")
2 sel0 = which("261892_at" == fData(gse25171)[, "PROBEID"])
3 df0 = data.frame(pData(gse25171)[, c("time", "Pi")],
4               expression=exprs(gse25171)[sel0,])
1 (p = ggplot(df0, aes(x=time, y=expression)) + geom_point())
```

- Podemos querer guardar el gráfico.

```
1 ggsave("gse25171_261892_at.png",p)
```

## Datos

Expression vs time

## Recta de mínimos cuadrados

- Ajustamos.

```
1 fit = lm(expression ~ time, data = df0)
```

- ¿Qué clase de objeto tenemos?

```
1 class(fit)
```

```
[1] "lm"
```

- Vemos lo que el método `print` nos muestra de un objeto de clase `lm`.

```
1 fit
```

Call:

```
lm(formula = expression ~ time, data = df0)
```

Coefficients:

```
(Intercept)      time
    7.9684      -0.1331
```

- Los coeficientes de la recta de mínimos cuadrados los podemos obtener con el método `coef` o `coefficients`.

```
1 coef(fit)
```

```
(Intercept)      time
    7.9684362  -0.1330703
```

```
1 coefficients(fit)
```

```
(Intercept)      time
    7.9684362  -0.1330703
```

- Añadimos la recta de mínimos cuadrados al diagrama de puntos.

```
1 (p = ggplot(df0,aes(x=time,y=expression))+geom_point() +
2   geom_smooth(method='lm', se = FALSE))
```

- El método `predict` aplicado al objeto de clase `lm` nos proporciona las predicciones para las observaciones.

```
1 predict(fit)
```

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| GSM618324.CEL.gz | GSM618325.CEL.gz | GSM618326.CEL.gz | GSM618327.CEL.gz |
| 7.968436         | 7.968436         | 7.835366         | 7.835366         |
| GSM618328.CEL.gz | GSM618329.CEL.gz | GSM618330.CEL.gz | GSM618331.CEL.gz |
| 7.170014         | 7.170014         | 4.774749         | 4.774749         |
| GSM618332.CEL.gz | GSM618333.CEL.gz | GSM618334.CEL.gz | GSM618335.CEL.gz |
| 7.968436         | 7.968436         | 7.835366         | 7.835366         |
| GSM618336.CEL.gz | GSM618337.CEL.gz | GSM618338.CEL.gz | GSM618339.CEL.gz |
| 7.170014         | 7.170014         | 4.774749         | 4.774749         |
| GSM618340.CEL.gz | GSM618341.CEL.gz | GSM618342.CEL.gz | GSM618343.CEL.gz |
| 7.968436         | 7.968436         | 7.835366         | 7.835366         |
| GSM618344.CEL.gz | GSM618345.CEL.gz | GSM618346.CEL.gz | GSM618347.CEL.gz |
| 7.170014         | 7.170014         | 4.774749         | 4.774749         |

- El método residuals o resid nos proporciona los residuos.

```
1 residuals(fit)
```

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| GSM618324.CEL.gz | GSM618325.CEL.gz | GSM618326.CEL.gz | GSM618327.CEL.gz |
| 0.3545741        | 0.8777147        | -0.9836174       | 0.5831095        |
| GSM618328.CEL.gz | GSM618329.CEL.gz | GSM618330.CEL.gz | GSM618331.CEL.gz |
| -1.2520196       | 0.9791937        | -0.4345729       | 0.3560575        |
| GSM618332.CEL.gz | GSM618333.CEL.gz | GSM618334.CEL.gz | GSM618335.CEL.gz |
| -0.1650657       | 0.5102758        | -1.2935869       | 0.8132667        |
| GSM618336.CEL.gz | GSM618337.CEL.gz | GSM618338.CEL.gz | GSM618339.CEL.gz |
| -1.6008608       | 1.1191170        | -0.4417852       | -0.2617990       |
| GSM618340.CEL.gz | GSM618341.CEL.gz | GSM618342.CEL.gz | GSM618343.CEL.gz |
| -0.4191948       | 0.8119677        | -1.1244602       | 1.1839258        |
| GSM618344.CEL.gz | GSM618345.CEL.gz | GSM618346.CEL.gz | GSM618347.CEL.gz |
| -1.0545087       | 0.2315680        | 0.4836551        | 0.7270455        |

```
1 resid(fit)
```

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| GSM618324.CEL.gz | GSM618325.CEL.gz | GSM618326.CEL.gz | GSM618327.CEL.gz |
| 0.3545741        | 0.8777147        | -0.9836174       | 0.5831095        |
| GSM618328.CEL.gz | GSM618329.CEL.gz | GSM618330.CEL.gz | GSM618331.CEL.gz |
| -1.2520196       | 0.9791937        | -0.4345729       | 0.3560575        |
| GSM618332.CEL.gz | GSM618333.CEL.gz | GSM618334.CEL.gz | GSM618335.CEL.gz |
| -0.1650657       | 0.5102758        | -1.2935869       | 0.8132667        |
| GSM618336.CEL.gz | GSM618337.CEL.gz | GSM618338.CEL.gz | GSM618339.CEL.gz |
| -1.6008608       | 1.1191170        | -0.4417852       | -0.2617990       |
| GSM618340.CEL.gz | GSM618341.CEL.gz | GSM618342.CEL.gz | GSM618343.CEL.gz |
| -0.4191948       | 0.8119677        | -1.1244602       | 1.1839258        |
| GSM618344.CEL.gz | GSM618345.CEL.gz | GSM618346.CEL.gz | GSM618347.CEL.gz |
| -1.0545087       | 0.2315680        | 0.4836551        | 0.7270455        |

- El método summary aplicado a un objeto lm.

```
1 (fit.s = summary(fit))
```

Call:

```
lm(formula = expression ~ time, data = df0)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-1.6009 -0.5772  0.2931  0.7483  1.1839
```

```
Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.96844    0.23130  34.451 < 2e-16 ***
time        -0.13307    0.01868  -7.122 3.84e-07 ***
---
```

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

```
Residual standard error: 0.8836 on 22 degrees of freedom
Multiple R-squared:  0.6975,    Adjusted R-squared:  0.6837
F-statistic: 50.73 on 1 and 22 DF,  p-value: 3.842e-07
```

- Tenemos un objeto de clase

```
1 class(fit.s)
```

```
[1] "summary.lm"
```

- Este resumen tiene muchas cosas como podemos ver

```
1 str(fit.s)
```

```
List of 11
```

```
$ call      : language lm(formula = expression ~ time, data = df0)
$ terms     :Classes 'terms', 'formula' language expression ~ time
.. ..- attr(*, "variables")= language list(expression, time)
.. ..- attr(*, "factors")= int [1:2, 1] 0 1
.. .. ..- attr(*, "dimnames")=List of 2
.. .. .. ..$ : chr [1:2] "expression" "time"
.. .. .. ..$ : chr "time"
.. ..- attr(*, "term.labels")= chr "time"
.. ..- attr(*, "order")= int 1
.. ..- attr(*, "intercept")= int 1
.. ..- attr(*, "response")= int 1
.. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
.. ..- attr(*, "predvars")= language list(expression, time)
.. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
.. .. ..- attr(*, "names")= chr [1:2] "expression" "time"
$ residuals : Named num [1:24] 0.355 0.878 -0.984 0.583 -1.252 ...
..- attr(*, "names")= chr [1:24] "GSM618324.CEL.gz" "GSM618325.CEL.gz" "GSM618326.CEL.gz"
$ coefficients : num [1:2, 1:4] 7.9684 -0.1331 0.2313 0.0187 34.4513 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : chr [1:2] "(Intercept)" "time"
.. ..$ : chr [1:4] "Estimate" "Std. Error" "t value" "Pr(>|t|)"
```

```

$ aliased      : Named logi [1:2] FALSE FALSE
..- attr(*, "names")= chr [1:2] "(Intercept)" "time"
$ sigma       : num 0.884
$ df          : int [1:3] 2 22 2
$ r.squared   : num 0.697
$ adj.r.squared: num 0.684
$ fstatistic  : Named num [1:3] 50.7 1 22
..- attr(*, "names")= chr [1:3] "value" "numdf" "dendf"
$ cov.unscaled : num [1:2, 1:2] 0.068522 -0.003465 -0.003465 0.000447
..- attr(*, "dimnames")=List of 2
.. ..$ : chr [1:2] "(Intercept)" "time"
.. ..$ : chr [1:2] "(Intercept)" "time"
- attr(*, "class")= chr "summary.lm"

```

- Una versión simplificada puede ser

```
1 attributes(fit.s)
```

```

$names
 [1] "call"          "terms"          "residuals"      "coefficients"
 [5] "aliased"       "sigma"          "df"              "r.squared"
 [9] "adj.r.squared" "fstatistic"    "cov.unscaled"

```

```

$class
 [1] "summary.lm"

```

- En particular el coeficiente de determinación lo tenemos con

```
1 fit.s$r.squared
```

```
[1] 0.6974934
```

- La matriz de modelo se obtiene con

```
1 model.matrix(fit)
```

```

              (Intercept) time
GSM618324.CEL.gz         1    0
GSM618325.CEL.gz         1    0
GSM618326.CEL.gz         1    1
GSM618327.CEL.gz         1    1
GSM618328.CEL.gz         1    6
GSM618329.CEL.gz         1    6
GSM618330.CEL.gz         1   24
GSM618331.CEL.gz         1   24
GSM618332.CEL.gz         1    0
GSM618333.CEL.gz         1    0
GSM618334.CEL.gz         1    1
GSM618335.CEL.gz         1    1
GSM618336.CEL.gz         1    6

```

```

GSM618337.CEL.gz      1    6
GSM618338.CEL.gz      1   24
GSM618339.CEL.gz      1   24
GSM618340.CEL.gz      1    0
GSM618341.CEL.gz      1    0
GSM618342.CEL.gz      1    1
GSM618343.CEL.gz      1    1
GSM618344.CEL.gz      1    6
GSM618345.CEL.gz      1    6
GSM618346.CEL.gz      1   24
GSM618347.CEL.gz      1   24
attr(,"assign")
[1] 0 1

```

## Contrastes e intervalos para la pendiente

- En el summary tenemos los contrastes.

```

1 summary(fit)

Call:
lm(formula = expression ~ time, data = df0)

Residuals:
    Min       1Q   Median       3Q      Max
-1.6009 -0.5772  0.2931  0.7483  1.1839

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.96844    0.23130  34.451 < 2e-16 ***
time         -0.13307    0.01868  -7.122 3.84e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8836 on 22 degrees of freedom
Multiple R-squared:  0.6975,    Adjusted R-squared:  0.6837
F-statistic: 50.73 on 1 and 22 DF,  p-value: 3.842e-07

```

- Los intervalos de confianza con nivel 0.95 vienen con

```

1 confint(fit)

              2.5 %      97.5 %
(Intercept)  7.4887585  8.4481140
time         -0.1718183 -0.0943223

```