

Aplicaciones del modelo de regresión lineal: cálculo de predicciones

Contents

1	Aplicaciones de la regresión	1
2	Predicción del valor medio	1
3	Intervalo de predicción	2
4	Conclusiones	3
5	Ejemplo	3
5.1	Predicción en un modelo de regresión simple	4
5.2	Predicción en un modelo de regresión múltiple	6
6	Predicciones utilizando bootstrap	7
6.1	Intervalo de confianza para el valor medio	7
6.2	Intervalo de predicción	8

1 Aplicaciones de la regresión

Podemos identificar dos aplicaciones básicas de los modelos de regresión:

- Predecir.
- Describir relaciones entre variables.

2 Predicción del valor medio

Sea el modelo de regresión

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + u_i, \quad i = 1, 2, \dots, n$$

Que se también se puede escribir como:

$$y_i = \begin{bmatrix} 1 & x_{1i} & x_{2i} & \cdots & x_{ki} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} + u_i = x_i^T \beta + u_i, \quad i = 1, 2, \dots, n$$

Utilizando parámetros estimados y residuos:

$$y_i = \begin{bmatrix} 1 & x_{1i} & x_{2i} & \cdots & x_{ki} \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_k \end{bmatrix} + u_i = x_i^T \hat{\beta} + e_i, \quad i = 1, 2, \dots, n$$

Dado un valor “nuevo” de los regresores $x_p^T = [1 \ x_{1p} \ x_{2p} \ \cdots \ x_{kp}]$, se define la predicción del valor medio y en x_p como:

$$\hat{y}_p = x_p^T \hat{\beta}$$

Es un estimador centrado del valor medio de y_p , $E[y_p]$:

$$E[\hat{y}_p] = E[x_p^T \hat{\beta}] = x_p^T E[\hat{\beta}] = x_p^T \beta$$

Y con varianza:

$$Var[\hat{y}_p] = Var[x_p^T \hat{\beta}] = x_p^T Var[\hat{\beta}] x_p = \sigma^2 x_p^T (X^T X)^{-1} x_p$$

Como $\hat{\beta}$ tiene distribución normal, se tiene que:

$$\hat{y}_p \sim N(x_p^T \beta, \sigma^2 x_p^T (X^T X)^{-1} x_p) \Rightarrow \frac{\hat{y}_p - x_p^T \beta}{\hat{s}_R \sqrt{x_p^T (X^T X)^{-1} x_p}} \sim t_{n-k-1}$$

Finalmente, el intervalo de confianza para $E[y_p] = x_p^T \beta$ es:

$$x_p^T \hat{\beta} - t_{\alpha/2} \hat{s}_R \sqrt{x_p^T (X^T X)^{-1} x_p} \leq (x_p^T \beta) \leq x_p^T \hat{\beta} + t_{\alpha/2} \hat{s}_R \sqrt{x_p^T (X^T X)^{-1} x_p}$$

Recordad que los intervalos de confianza se definen para parámetros del modelo. En este caso el intervalo de confianza se ha definido para una combinación lineal de parámetros.

3 Intervalo de predicción

El valor real para x_p es y_p y no $x_p^T \hat{\beta}$. Por tanto, si utilizamos como predicción de y_p el valor $\hat{y}_p = x_p^T \hat{\beta}$ tenemos un error de predicción:

$$\epsilon_p = y_p - \hat{y}_p = y_p - x_p^T \hat{\beta}$$

La variable aleatoria ϵ_p no es un residuo. Los residuos se definen para datos observados. Tenemos que suponer que y_p verifica la ecuación del modelo, esto es, $y_p = x_p^T \beta + u_p$, donde $u_p \sim N(0, \sigma^2)$. Entonces

$$y_p \sim N(x_p^T \beta, \sigma^2)$$

Nos gustaría construir un intervalo (a, b) para la y_p tal que:

$$P(a \leq y_p \leq b) = 1 - \alpha$$

Sin embargo no podemos utilizar la distribución de y_p ya que desconocemos β y σ^2 .

Por otro lado, acabamos de encontrar que

$$x_p^T \hat{\beta} \sim N(x_p^T \beta, \sigma^2 x_p^T (X^T X)^{-1} x_p)$$

Por tanto podemos averiguar la distribución de ϵ_p :

$$\epsilon_p = y_p - x_p^T \hat{\beta}$$

$$\epsilon_p \sim N(0, \sigma^2(1 + x_p^T(X^T X)^{-1}x_p))$$

ya que:

$$E[\epsilon_p] = E[y_p] - x_p^T E[\hat{\beta}] = x_p^T \beta - x_p^T \beta = 0$$

$$Var[\epsilon_p] = Var[y_p] + Var[x_p^T \hat{\beta}] = \sigma^2 + x_p^T Var[\hat{\beta}] x_p = \sigma^2(1 + x_p^T(X^T X)^{-1}x_p)$$

donde se ha considerado que y_p e \hat{y}_p son independientes. Utilizando la varianza residual tenemos:

$$\frac{\epsilon_p}{\hat{s}_R \sqrt{1 + x_p^T(X^T X)^{-1}x_p}} \sim t_{n-k-1}$$

Con lo que se puede encontrar que:

$$P\left(-t_{\alpha/2} \hat{s}_R \sqrt{1 + x_p^T(X^T X)^{-1}x_p} \leq \epsilon_p \leq t_{\alpha/2} \hat{s}_R \sqrt{1 + x_p^T(X^T X)^{-1}x_p}\right) = 1 - \alpha$$

Finalmente, el intervalo para y_p que estábamos buscando se calcula como:

$$P\left(x_p^T \hat{\beta} - t_{\alpha/2} \hat{s}_R \sqrt{1 + x_p^T(X^T X)^{-1}x_p} \leq y_p \leq x_p^T \hat{\beta} + t_{\alpha/2} \hat{s}_R \sqrt{1 + x_p^T(X^T X)^{-1}x_p}\right) = 1 - \alpha$$

Esto no es un intervalo de confianza ya que y_p no es un parámetro, sino que es un intervalo de probabilidad (y_p es una variable aleatoria).

4 Conclusiones

Sea un valor para los regresores x_p . Según el modelo $y_p = x_p^T \beta + u_p$, $u_p \sim N(0, \sigma^2)$:

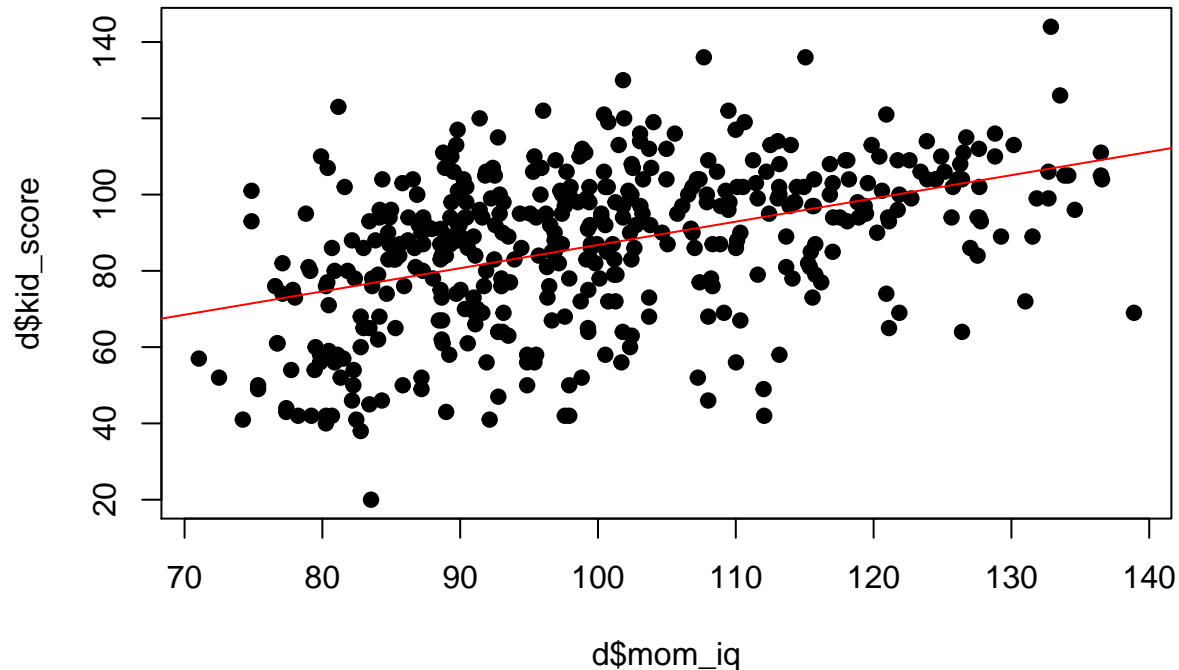
- Si queremos asignar un *valor puntual* para la predicción de y_p utilizaremos $\hat{y}_p = x_p^T \hat{\beta}$, que es un estimador centrado de $x_p^T \beta$.
- Si queremos construir un intervalo para la predicción en el punto x_p utilizaremos el intervalo de y_p .

5 Ejemplo

```
d = read.csv("datos/kidiq.csv")
d$mom_hs = factor(d$mom_hs, labels = c("no", "si"))
d$mom_work = factor(d$mom_work, labels = c("notrabaja", "trabaja23", "trabaja1_parcial", "trabaja1_comp"))
```

5.1 Predicción en un modelo de regresión simple

```
m = lm(kid_score ~ mom_iq, data = d)
plot(d$mom_iq, d$kid_score, pch = 19)
abline(m, col = "red", lwd = 1)
```



5.1.1 Predicción del valor medio

```
xp = matrix(c(1, 130), ncol = 1)
n = nrow(d)
beta_e = coef(m)
sR2 = sum(resid(m)^2)/(n-2)
X = model.matrix(m)
mat = t(xp) %*% solve(t(X) %*% X) %*% xp
# predicción puntual
(yp_medio = t(xp) %*% beta_e)

##           [,1]
## [1,] 105.0965

# intervalo de confianza
yp_medio[1,1] + c(-1,1)*qt(0.975,n-2)*sqrt(sR2*(mat[1,1]))

## [1] 101.2394 108.9535
```

En R:

```
xp1 = data.frame(mom_iq = 130)
(yp_medio1 = predict(m, newdata = xp1, interval = "confidence", level = 0.95))

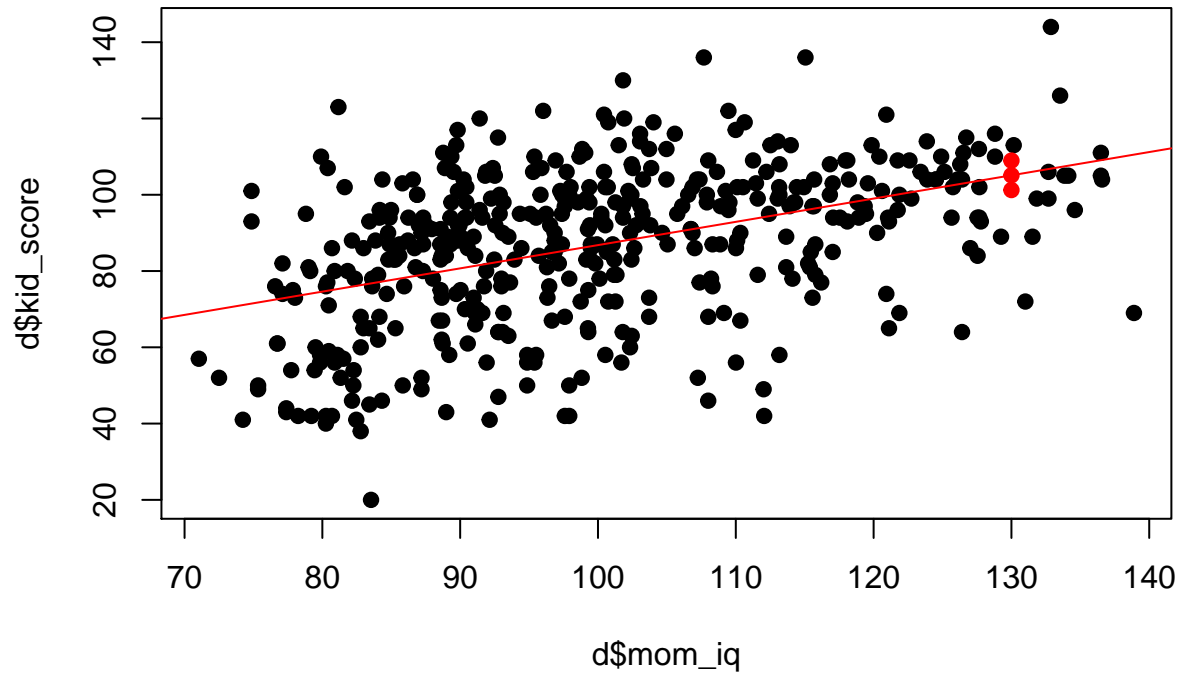
##          fit      lwr      upr
## 1 105.0965 101.2394 108.9535

plot(d$mom_iq, d$kid_score, pch = 19)
abline(m, col = "red", lwd = 1)
```

```

points(xp1$mom_iq, yp_medio1[1], col = "red", pch = 19) # prediccion puntual
points(xp1$mom_iq, yp_medio1[2], col = "red", pch = 19) # limite inferior int. conf.
points(xp1$mom_iq, yp_medio1[3], col = "red", pch = 19) # limite superior int. conf.

```



5.1.2 Intervalo de prediccion

```

(yp = yp_medio[1,1] + c(-1,1)*qt(0.975,n-2)*sqrt(sR2*(1 + mat[1,1])))

```

```
## [1] 68.98835 141.20459
```

- En R:

```

(yp1 = predict(m, newdata = xp1, interval = "prediction", level = 0.95))

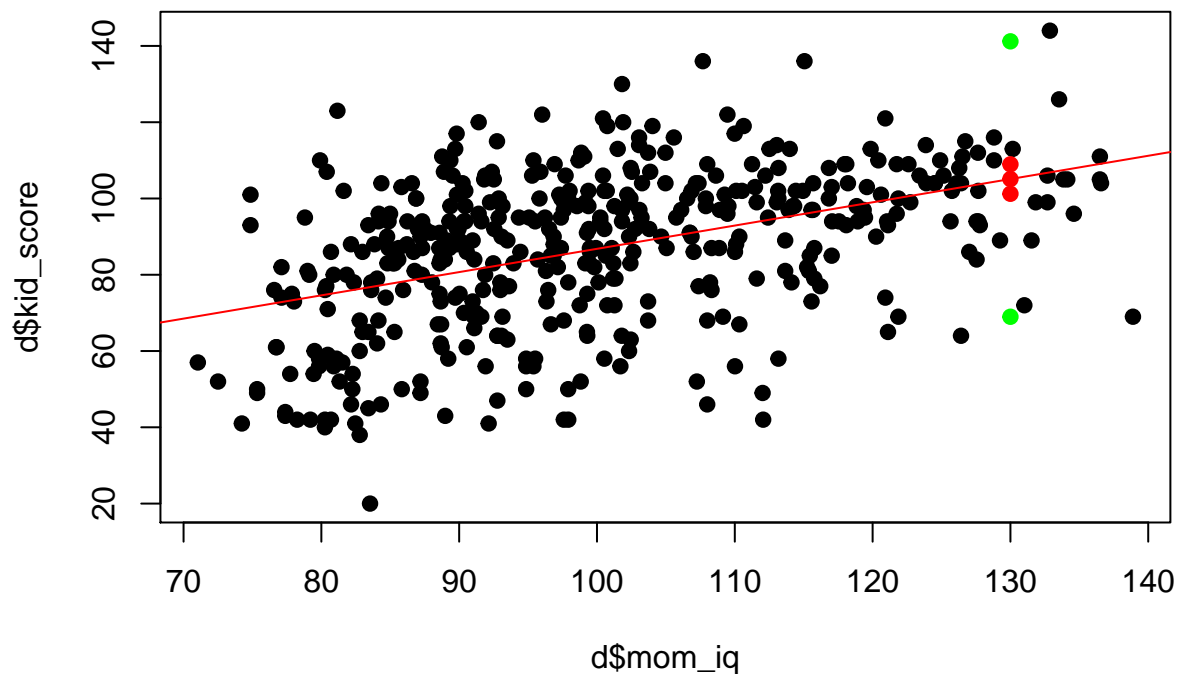
```

```
##          fit          lwr          upr
## 1 105.0965 68.98835 141.2046
```

```

plot(d$mom_iq, d$kid_score, pch = 19)
abline(m, col = "red", lwd = 1)
points(xp1$mom_iq, yp_medio1[1], col = "red", pch = 19) # prediccion puntual
points(xp1$mom_iq, yp_medio1[2], col = "red", pch = 19) # limite inferior int. conf.
points(xp1$mom_iq, yp_medio1[3], col = "red", pch = 19) # limite superior int. conf.
points(xp1$mom_iq, yp1[2], col = "green", pch = 19) # limite inferior int. pred.
points(xp1$mom_iq, yp1[3], col = "green", pch = 19) # limite superior int. pred.

```



5.2 Predicción en un modelo de regresión múltiple

Vamos a predecir:

- mom_iq = 130
- mom_hs = no
- mom_age = 25
- mom_work = trabaja1_parcial

```
m2 = lm(kid_score ~ mom_iq + mom_hs + mom_age + mom_work, data = d)
summary(m2)
```

```
##
## Call:
## lm(formula = kid_score ~ mom_iq + mom_hs + mom_age + mom_work,
##     data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -54.414 -12.095   2.015  11.653  49.100
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    20.27273    9.39320   2.158  0.0315 *
## mom_iq          0.55288    0.06138   9.008 <2e-16 ***
## mom_hssi        5.43466    2.32518   2.337  0.0199 *
## mom_age         0.21629    0.33351   0.649  0.5170
## mom_worktrabaja23 2.98266    2.81289   1.060  0.2896
## mom_worktrabaja1_parcial 5.48824    3.25239   1.687  0.0922 .
## mom_worktrabaja1_completo 1.41929    2.51621   0.564  0.5730
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 18.14 on 427 degrees of freedom
## Multiple R-squared:  0.2213, Adjusted R-squared:  0.2103
## F-statistic: 20.22 on 6 and 427 DF,  p-value: < 2.2e-16
```

5.2.1 Prediccion del valor medio

Recordamos que el modelo sería:

$$kid_score = \hat{\beta}_0 + \hat{\beta}_1 mom_iq + \hat{\beta}_2 mom_hssi + \hat{\beta}_3 mon_age + \hat{\beta}_4 mom_worktrabaja23 + \hat{\beta}_5 mom_worktrabaja1_parcial + \hat{\beta}_6 mom_worktrabaja1_parcial^2$$

```
xp = matrix(c(1, 130, 0, 25, 0, 1, 0), ncol = 1)
beta_e = coef(m2)
sR2 = sum(resid(m2)^2)/(n-6-1)
X = model.matrix(m2)
mat = t(xp) %*% solve(t(X) %*% X) %*% xp
# prediccion del valor medio
(yp_medio = t(xp) %*% beta_e)

##           [,1]
## [1,] 103.0423
# intervalo de confianza
yp_medio[1,1] + c(-1,1)*qt(0.975,n-6-1)*sqrt(sR2*(mat[1,1]))

## [1] 95.74381 110.34083
```

- En R:

```
xp1 = data.frame(mom_iq = 130, mom_hs = "no", mom_age = 25, mom_work = "trabaja1_parcial")
(yp_medio1 = predict(m2, newdata = xp1, interval = "confidence", level = 0.95))

##           fit      lwr      upr
## 1 103.0423 95.74381 110.3408
```

5.2.2 Intervalo de prediccion

```
(yp = yp_medio[1,1] + c(-1,1)*qt(0.975,n-6-1)*sqrt(sR2*(1 + mat[1,1])))

## [1] 66.65286 139.43178
```

- En R:

```
(yp1 = predict(m2, newdata = xp1, interval = "prediction", level = 0.95))

##           fit      lwr      upr
## 1 103.0423 66.65286 139.4318
```

6 Predicciones utilizando bootstrap

6.1 Intervalo de confianza para el valor medio

Vamos a calcular el intervalo de confianza utilizando bootstrap:

```
B = 1000
yp_medio_b = rep(0,B)
e = resid(m2)
for (b in 1:B){
```

```

eb = sample(e, replace = T)
yb = fitted(m2) + eb
mb = lm(yb ~ mom_iq + mom_hs + mom_age + mom_work, data = d)
yp_medio_b[b] = predict(mb, newdata = xp1, interval = "none", level = 0.95)
}

```

El intervalo de confianza para el valor medio es:

```
quantile(yp_medio_b, probs = c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 95.76144 110.15791
```

6.2 Intervalo de predicción

Vamos a calcular el intervalo de predicción utilizando la fórmula:

$$\epsilon_p = y_p - \hat{y}_p \Rightarrow y_p = \hat{y}_p + \epsilon_p$$

En este caso no se puede utilizar bootstrap. El método bootstrap se utiliza para calcular intervalos de confianza y se basa en remuestrear residuos. Ahora tenemos una variable aleatoria, no unos residuos. Podemos utilizar simulaciones de la variable aleatoria:

```

# la variable epsilon_p tiene varianza sR2*(1+mat)
sR2 = sum(resid(m2)^2)/(n-2)
epsilon_var = sR2*(1 + mat[1,1])
# simulamos la variable epsilon
epsilon = rnorm(1000, mean = 0, sd = sqrt(epsilon_var))
xp = matrix(c(1, 130, 0, 25, 0, 1, 0), ncol = 1)
yp = t(xp) %*% coef(m2)
yp_b = yp[1,1] + epsilon
# intervalo de prediccion
quantile(yp_b, probs = c(0.025, 0.975))

```

```
##      2.5%      97.5%
## 64.48184 139.36849
```

Aunque en este caso, al ser una distribución normal, lo más sencillo es:

```
qnorm(c(0.025,0.975), mean = yp, sd = sqrt(epsilon_var))
```

```
## [1] 66.96661 139.11804
```