

# Aplicaciones del modelo de regresión lineal: cálculo de predicciones

## Contents

<b>1 Aplicaciones de la regresión</b>	<b>1</b>
<b>2 Estimación del valor medio</b>	<b>1</b>
<b>3 Varianza de la estimación del valor medio</b>	<b>2</b>
3.1 Con matrices de datos . . . . .	2
3.2 Con matrices de covarianzas . . . . .	2
<b>4 Intervalo de confianza de la estimación del valor medio</b>	<b>3</b>
<b>5 Intervalo de predicción</b>	<b>3</b>
<b>6 Conclusiones</b>	<b>4</b>
<b>7 Ejemplo</b>	<b>4</b>
7.1 Predicción en un modelo de regresión simple . . . . .	5
7.2 Predicción en un modelo de regresión múltiple . . . . .	7
<b>8 Predicciones utilizando bootstrap</b>	<b>9</b>
8.1 Intervalo de confianza para el valor medio . . . . .	9
8.2 Intervalo de predicción . . . . .	9

## 1 Aplicaciones de la regresión

Podemos identificar dos aplicaciones básicas de los modelos de regresión:

- Predecir.
- Describir relaciones entre variables.

La predicción en el modelo de regresión se puede enfocar de dos formas diferentes

- Estimación del valor medio y su intervalo de confianza.
- Intervalo de predicción para un valor de  $y$ .

## 2 Estimación del valor medio

Sea el modelo de regresión

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

Este modelo se puede escribir como:

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

Por ejemplo, si consideramos los valores de los regresores  $x_p^T = [1 \ x_{1p} \ x_{2p} \ \cdots \ x_{kp}]$ , se tiene que

$$y_p = x_p^T \beta + u_p, \quad u_p \sim N(0, \sigma^2)$$

El valor medio de  $y_p$  es:

$$E[y_p] = E[x_p^T \beta + u_p] = x_p^T \beta$$

Por este motivo es usual utilizar el siguiente estimador de  $E[y_p]$ :

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

es decir, sustituimos  $\beta$  por sus estimadores. Este valor es un estimador centrado de  $E[y_p]$ :

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

y además tiene varianza mínima (se omite la demostración). En la práctica, cuando se habla de predecir el valor de  $y$  en el punto  $x_p$  lo que se hace es estimar el valor medio de  $y_p$ , es decir,  $\hat{y}_p$ .

### 3 Varianza de la estimación del valor medio

#### 3.1 Con matrices de datos

La varianza se calcula como:

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

donde

$$v_p = x_p^T (X^T X)^{-1} x_p$$

#### 3.2 Con matrices de covarianzas

Se tiene que

$$\hat{y}_p = x_p^T \hat{\beta} = \hat{\beta}_0 + [x_{1p} \ x_{2p} \ \cdots \ x_{kp}] \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_k \end{bmatrix} = \hat{\beta}_0 + \tilde{x}_p^T \hat{\beta}_a$$

donde  $\tilde{x}_p^T = [x_{1p} \ x_{2p} \ \cdots \ x_{kp}]$ . Por otro lado hemos visto que

$$\bar{y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x}_1 + \hat{\beta}_2 \bar{x}_2 + \cdots + \hat{\beta}_k \bar{x}_k = \hat{\beta}_0 + \bar{x}^T \hat{\beta}_a$$

donde  $\bar{x}^T = [\bar{x}_1 \ \bar{x}_2 \ \cdots \ \bar{x}_k]$ . Despejando  $\hat{\beta}_0$  y sustituyendo en la otra ecuación se obtiene

$$\hat{y}_p = \bar{y} + (\tilde{x}_p - \bar{x})^T \hat{\beta}_a$$

Por tanto:

$$Var[\hat{y}_p] = Var[\bar{y}] + (\tilde{x}_p - \bar{x})^T Var[\hat{\beta}_a] (\tilde{x}_p - \bar{x})$$

En los temas anteriores se ha visto que  $Var[\bar{y}] = \frac{\sigma^2}{n}$  y que  $Var[\hat{\beta}_a] = \frac{\sigma^2}{n-1} S_{XX}^{-1}$ . Por tanto:

$$Var[\hat{y}_p] = \frac{\sigma^2}{n} + \frac{\sigma^2}{n-1} (\tilde{x}_p - \bar{x})^T S_{XX}^{-1} (\tilde{x}_p - \bar{x}) = \sigma^2 v_p$$

donde en este caso

$$v_p = \frac{1}{n} + \frac{1}{n-1} (\tilde{x}_p - \bar{x})^T S_{XX}^{-1} (\tilde{x}_p - \bar{x})$$

## 4 Intervalo de confianza de la estimación del valor medio

Primero vamos a deducir la distribución de  $\hat{y}_p = x_p^T \hat{\beta}$ . Como  $\hat{\beta}$  tiene distribución normal, se tiene que:

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

donde  $se(\hat{y}_p) = \hat{s}_R \sqrt{v_p}$ . Finalmente, el intervalo de confianza para  $E[y_p] = x_p^T \beta$  es:

$$\hat{y}_p - t_{\alpha/2} se(\hat{y}_p) \leq E[y_p] \leq \hat{y}_p + t_{\alpha/2} se(\hat{y}_p)$$

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.

## 5 Intervalo de predicción

También podemos decir algo acerca de  $y_p$ , y no solo de su valor medio. Como  $y_p = x_p^T \beta + u_p$ , donde  $u_p \sim N(0, \sigma^2)$ , se tiene que:

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

Como  $y_p$  es una variable aleatoria, la predicción tiene que ser en forma de intervalo. Nos gustaría construir un intervalo  $(LI, LS)$  para  $y_p$  tal que:

$$P(LI \leq y_p \leq LS) = 1 - \alpha$$

Sin embargo no podemos utilizar la distribución de  $y_p$  ya que desconocemos  $\beta$  y  $\sigma^2$ . La opción es trabajar con la diferencia entre  $y_p$  y su valor medio predicho  $\hat{y}_p$ . Hemos visto que

$$\hat{y}_p \sim N(x_p^T \beta, \sigma^2 v_p)$$

Por tanto podemos calcular la distribución de  $y_p - \hat{y}_p$ :

$$y_p - \hat{y}_p \sim N(0, \sigma^2(1 + v_p))$$

ya que:

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

$$Var[y_p - \hat{y}_p] = Var[y_p] + Var[\hat{y}_p] = \sigma^2 + \sigma^2 v_p = \sigma^2(1 + v_p)$$

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

$$\frac{y_p - \hat{y}_p}{\hat{s}_R \sqrt{1 + v_p}} \sim t_{n-k-1}$$

Con lo que se puede encontrar que:

$$P(-t_{\alpha/2} \hat{s}_R \sqrt{1 + v_p} \leq y_p - \hat{y}_p \leq t_{\alpha/2} \hat{s}_R \sqrt{1 + v_p}) = 1 - \alpha$$

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

$$P(\hat{y}_p - t_{\alpha/2} \hat{s}_R \sqrt{1 + v_p} \leq y_p \leq \hat{y}_p + t_{\alpha/2} \hat{s}_R \sqrt{1 + v_p}) = 1 - \alpha$$

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).

## 6 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  $y_p$  es usual utilizar la estimación del valor medio,  $\hat{y}_p = x_p^T \hat{\beta}$ , que es un estimador centrado del valor medio teórico,  $x_p^T \beta$ . Es posible calcular un intervalo de confianza para el valor medio teórico.
- También se puede construir un intervalo para la predicción de  $y_p$ .

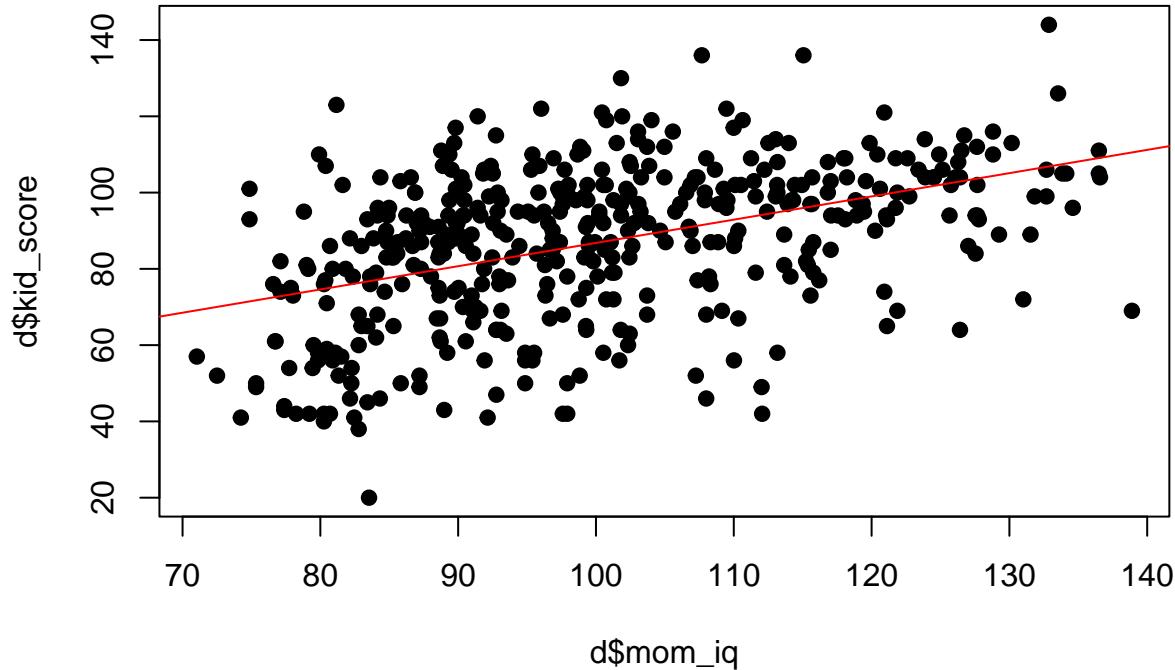
## 7 Ejemplo

```
load("datos/kidiq.Rdata")
str(d)

## 'data.frame':    434 obs. of  5 variables:
##   $ kid_score: int  65 98 85 83 115 98 69 106 102 95 ...
##   $ mom_hs   : Factor w/ 2 levels "no","si": 2 2 2 2 2 1 2 2 2 2 ...
##   $ mom_iq   : num  121.1 89.4 115.4 99.4 92.7 ...
##   $ mom_work : Factor w/ 4 levels "notrabajo","trabaja23",...: 4 4 4 3 4 1 4 3 1 1 ...
##   $ mom_age  : int  27 25 27 25 27 18 20 23 24 19 ...
```

## 7.1 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)
```



### 7.1.1 Estimación 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)
(vp = t(xp) %*% solve(t(X) %*% X) %*% xp)

## [,1]
## [1,] 0.01154202
# 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*(vp[1,1]))

## [1] 101.2394 108.9535
# para comprobar, vamos a calcular vp con la matriz de covarianzas
Sxx = var(d$mom_iq)
xp1 = 130
xm = mean(d$mom_iq)
(vp1 = 1/n + 1/(n-1)*(xp1 - xm)^2*1/Sxx)
```

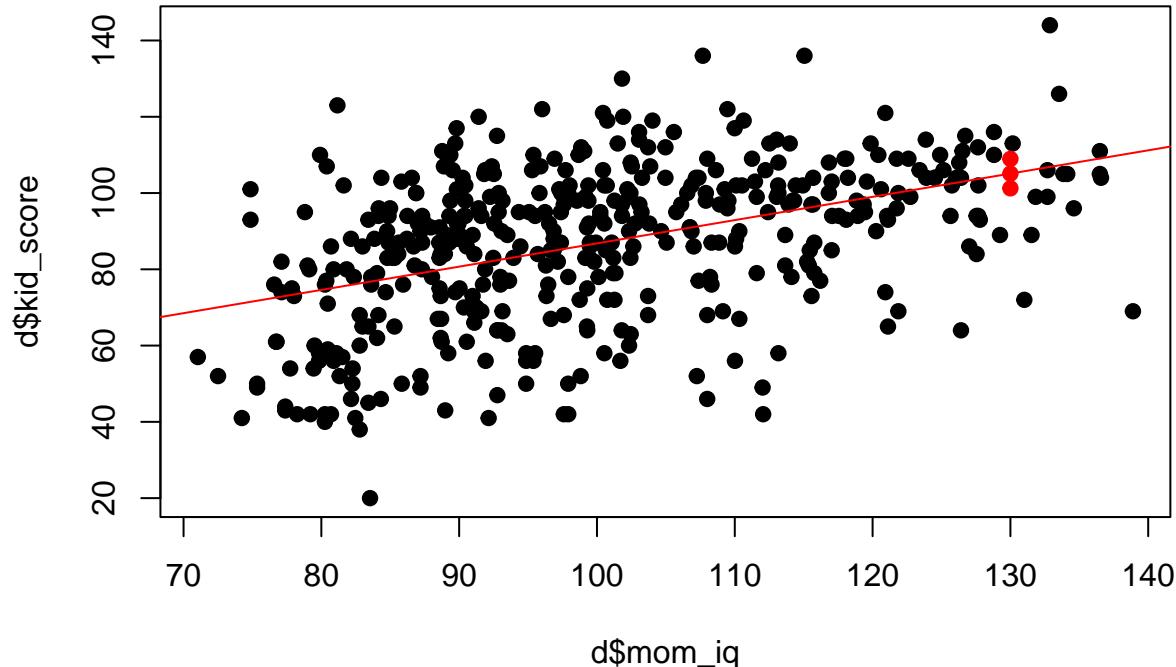
```
## [1] 0.01154202
```

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.
```



### 7.1.2 Intervalo de predicción

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

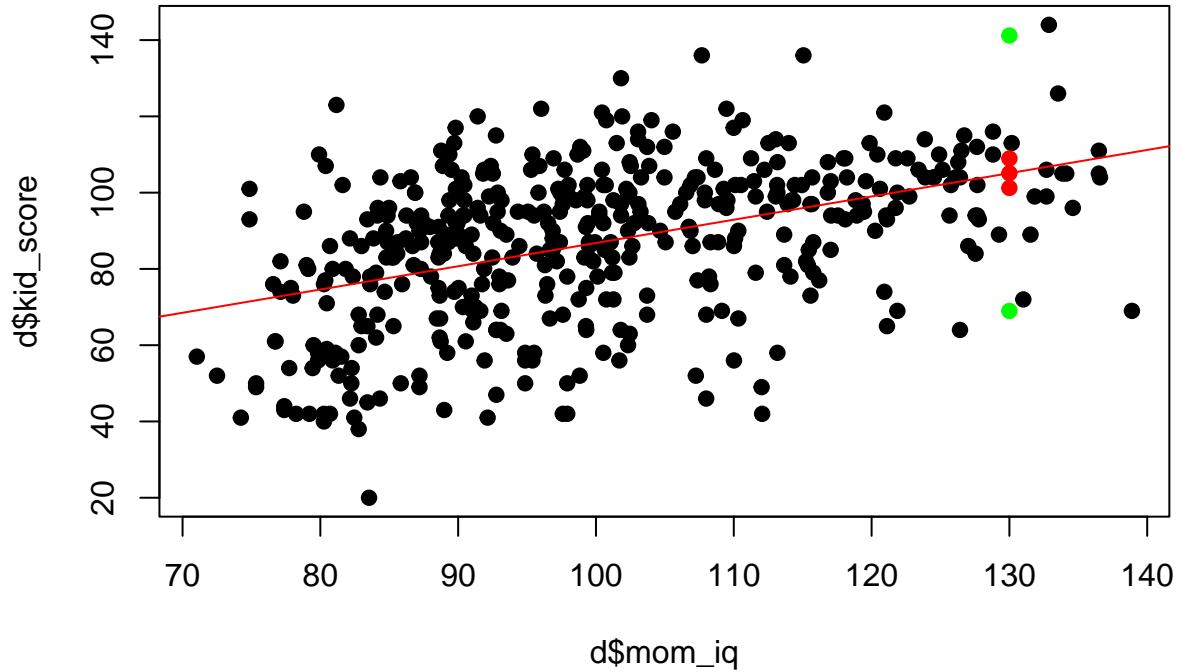
```
## [1] 68.98835 141.20459
```

- 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.
```



## 7.2 Predicción en un modelo de regresión múltiple

Vamos a predecir:

- mom\_iq = 130
- mon\_hs = no
- mom\_age = 25
- mom\_work = trabaja1p

```
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_worktrabaja1p 5.48824  3.25239  1.687   0.0922 .  
## mom_worktrabaja1c 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

```

### 7.2.1 Estimación del valor medio

Recordamos que el modelo sería:

$$kid\_score = \hat{\beta}_0 + \hat{\beta}_1 mom\_iq + \hat{\beta}_2 mom\_hs + \hat{\beta}_3 mom\_age + \hat{\beta}_4 mom\_worktrabaja23 + \hat{\beta}_5 mom\_worktrabaja1p + \hat{\beta}_6 mom\_wor$$

```

xp = matrix(c(1, 130, 0, 25, 0, 1, 0), ncol = 1)
beta_e = coef(m2)
k = 6 # numero de regresores
sR2 = sum(resid(m2)^2)/(n-k-1)
X = model.matrix(m2)
(vp = t(xp) %*% solve(t(X) %*% X) %*% xp)

## [1] [,1]
## [1,] 0.04191298

# predicción del valor medio
(yp_medio = t(xp) %*% beta_e)

## [1] [,1]
## [1,] 103.0423

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

## [1] 95.74381 110.34083

# para comprobar, vamos a calcular vp con la matriz de covarianzas
X1 = X[,2:(k+1)]
Sxx = var(X1)
xp1 = xp[2:(k+1),]
xm = apply(X1, 2, mean)
(vp1 = 1/n + 1/(n-1)*t(xp1 - xm) %*% solve(Sxx) %*% (xp1 - xm) )

## [1] [,1]
## [1,] 0.04191298

```

- En R:

```

xp1 = data.frame(mom_iq = 130, mom_hs = "no", mom_age = 25, mom_work = "trabaja1p")
(yp_medio1 = predict(m2, newdata = xp1, interval = "confidence", level = 0.95))

## fit      lwr      upr
## 1 103.0423 95.74381 110.3408

```

### 7.2.2 Intervalo de predicción

```

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

## 8 Predicciones utilizando bootstrap

### 8.1 Intervalo de confianza para el valor medio

Vamos a calcular el intervalo de confianza utilizando bootstrap:

```
n = nrow(d)
B = 1000
yp_medio_b = rep(0,B)
for (b in 1:B){
  pos = sample(1:n, replace = T)
  db = d[pos,]
  mb = lm(kid_score ~ mom_iq + mom_hs + mom_age + mom_work, data = db)
  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%
## 96.46424 109.21006
```

### 8.2 Intervalo de predicción

En este caso no se puede utilizar bootstrap. Con el modelo estimado se puede calcular  $\hat{y}_p = x_p^T \hat{\beta}$  en cada réplica de bootstrap; sin embargo no se puede calcular  $y_p = x_p^T \hat{\beta} + e_p$ , ya que  $e_p$  es desconocido (recordamos que  $e_p$  se define como  $y_p - \hat{y}_p$ , pero  $y_p$  es desconocido, es lo que queremos calcular). Por tanto, no es posible construir el intervalo de predicción con bootstrap.