

# Inferencia en el modelo de regresión lineal: intervalos de confianza

## Contents

<b>1</b>	<b>Introduccion</b>	<b>1</b>
<b>2</b>	<b>Intervalo de confianza para las <math>\beta_i</math></b>	<b>1</b>
2.1	Con matrices de datos . . . . .	1
2.2	Con matrices de covarianzas . . . . .	2
<b>3</b>	<b>Intervalo de confianza para <math>\sigma^2</math></b>	<b>3</b>
<b>4</b>	<b>Ejemplo</b>	<b>4</b>

## 1 Introduccion

Un intervalo de confianza para un parámetro es **un rango de valores posibles para dicho parámetro**.

## 2 Intervalo de confianza para las $\beta_i$

### 2.1 Con matrices de datos

Hemos visto que

$$\hat{\beta} \rightarrow N(\beta, \sigma^2 Q)$$

donde  $Q = (X^T X)^{-1}$ . Esto implica que:

$$\hat{\beta}_i \rightarrow N(\beta_i, \sigma^2 Q_{i+1, i+1}), \quad i = 0, 1, 2, \dots, k$$

donde  $Q_{i,i}$  es el elemento  $(i,i)$  de la matriz  $Q$ . Aplicando las propiedades de la distribución normal

$$\frac{\hat{\beta}_i - \beta_i}{\sqrt{\sigma^2 Q_{i+1, i+1}}} \rightarrow N(0, 1), \quad i = 0, 1, 2, \dots, k.$$

Por tanto:

$$\frac{\hat{\beta}_i - \beta_i}{se(\hat{\beta}_i)} \rightarrow t_{n-k-1}, \quad i = 0, 1, 2, \dots, k.$$

donde

$$se(\hat{\beta}_i) = \sqrt{\hat{s}_R^2 Q_{i+1, i+1}}, \quad i = 0, 1, 2, \dots, k.$$

Para deducir la expresión anterior se ha tenido en cuenta que

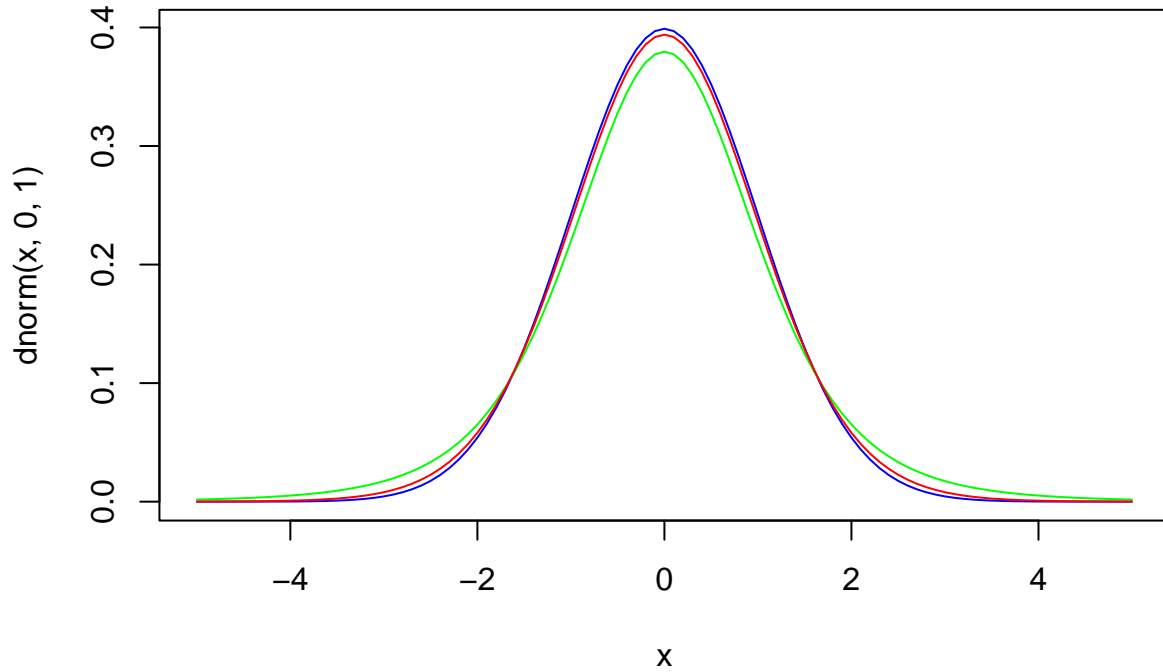
$$\frac{N(0,1)}{\sqrt{\frac{\chi_n^2}{n}}} \rightarrow t_n$$

Por tanto, el intervalo de confianza  $100(1 - \alpha)\%$  se obtiene como

$$\hat{\beta}_i \pm t_{n-k-1;\alpha/2} se(\hat{\beta}_i), \quad i = 0, 1, 2, \dots, k.$$

La distribución *t-student* es similar a la  $N(0,1)$ . De hecho,  $\lim_{n \rightarrow \infty} t_n = N(0,1)$ .

```
curve(dnorm(x,0,1), from = -5, to = 5, col = "blue")
curve(dt(x,5), add = T, col = "green")
curve(dt(x,20), add = T, col = "red")
```



## 2.2 Con matrices de covarianzas

Tenemos que

$$\hat{\beta}_a \rightarrow N(\beta_a, \sigma^2 Q_a)$$

donde

$$Q_a = \frac{1}{n-1} S_{XX}^{-1}$$

Esto implica que:

$$\hat{\beta}_i \rightarrow N(\beta_i, \sigma^2 Q_{a(i,i)}), \quad i = 1, 2, \dots, k$$

donde  $Q_{a(i,j)}$  es el elemento  $(i,j)$  de la matriz  $Q_a$ . Por tanto, siguiendo el razonamiento del apartado anterior:

$$se(\hat{\beta}_i) = \sqrt{\hat{s}_R^2 Q_{a(i,i)}}, \quad i = 1, 2, \dots, k$$

Para  $\hat{\beta}_0$  tenemos que

$$\hat{\beta}_0 \sim N\left(\beta_0, \sigma^2 \left(\frac{1}{n} + \frac{1}{n-1} \bar{x}^T S_{XX}^{-1} \bar{x}\right)\right)$$

Por tanto

$$se(\hat{\beta}_0) = \sqrt{\hat{s}_R^2 \left(\frac{1}{n} + \frac{1}{n-1} \bar{x}^T S_{XX}^{-1} \bar{x}\right)}$$

Finalmente, el intervalo de confianza  $100(1 - \alpha)\%$  se obtiene como

$$\hat{\beta}_i \pm t_{n-k-1; \alpha/2} se(\hat{\beta}_i), \quad i = 0, 1, 2, \dots, k.$$

### 3 Intervalo de confianza para $\sigma^2$

Partimos de la distribución en el muestreo:

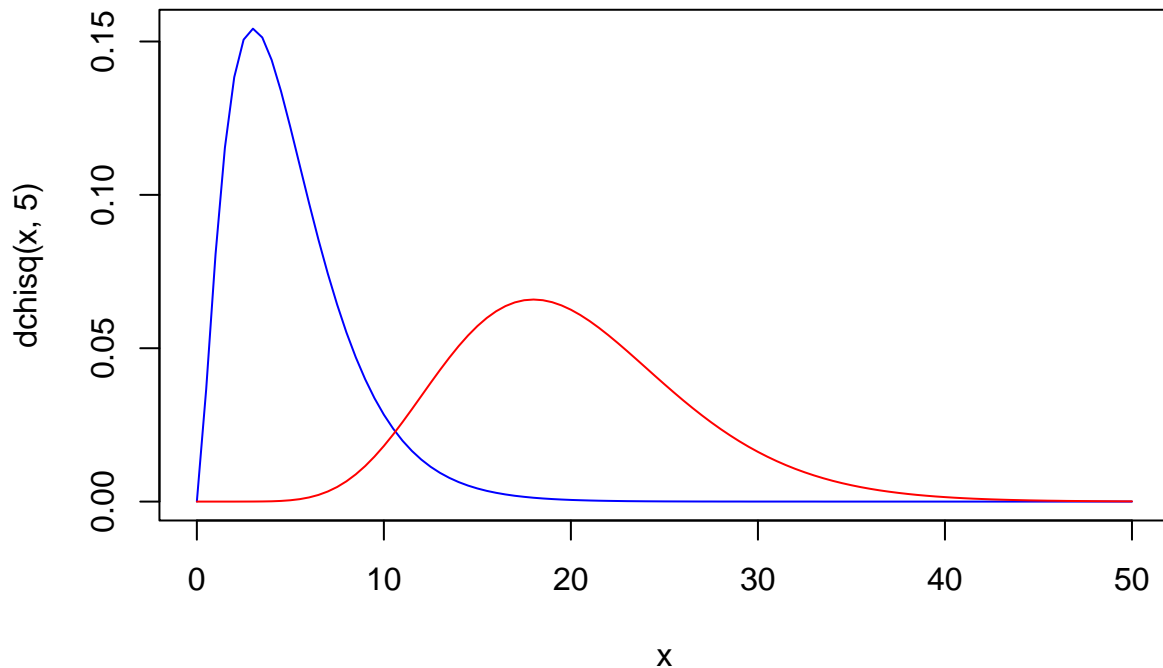
$$\frac{(n-k-1)\hat{s}_R^2}{\sigma^2} \rightarrow \chi_{n-k-1}^2$$

Despejando:

$$\frac{(n-k-1)\hat{s}_R^2}{\chi_{n-k-1; \alpha/2}^2} \leq \sigma^2 \leq \frac{(n-k-1)\hat{s}_R^2}{\chi_{n-k-1; 1-\alpha/2}^2}$$

Podemos dibujar la distribución  $\chi^2$ :

```
curve(dchisq(x,5), from = 0, to =50, col = "blue")
curve(dchisq(x,20), add = T, col = "red")
```



## 4 Ejemplo

Vamos a calcular de manera detallada los intervalos de confianza para el modelo  $kid\_score \sim mom\_iq + mom\_hs$ :

```
d = read.csv("datos/kidiq.csv")
#d$mom_hs = factor(d$mom_hs, labels = c("no", "si"))

m = lm(kid_score ~ mom_iq + mom_hs, data = d)
```

Los parámetros estimados son:

```
coef(m)

## (Intercept)      mom_iq      mom_hs
## 25.731538    0.563906    5.950117

# varianza residual
n = nrow(d)
k = 2 # numero de regresores
(sR2 = sum(resid(m)^2)/(n-k-1))

## [1] 328.9028
```

Vamos a calcular la varianza de los parámetros estimados, es decir  $var(\hat{\beta}_i) = \hat{s}_R^2 Q_{i+1,i+1}$ :

```
X = cbind(rep(1,n), d$mom_iq, d$mom_hs) # tambien X = model.matrix(m)
# Q = inv(t(X)*X)
(Q = solve(crossprod(X))) # crossprod es otra manera de calcular t(X) %*% X

##           [,1]      [,2]      [,3]
## [1,] 0.1049491626 -1.025110e-03 -0.0001705848
## [2,] -0.0010251098 1.115594e-05 -0.0001151616
## [3,] -0.0001705848 -1.151616e-04 0.0148740410
```

Por tanto, la matriz de varianzas de los estimadores será

```
(beta_var = sR2 * Q)
```

```
##           [,1]           [,2]           [,3]
## [1,] 34.51806922 -0.337161456 -0.05610582
## [2,] -0.33716146  0.003669219 -0.03787697
## [3,] -0.05610582 -0.037876974  4.89211314
```

Y el standard error de los estimadores,  $se(\hat{\beta}_i)$ :

```
(beta_se = sqrt(diag(beta_var)))
```

```
## [1] 5.87520802 0.06057408 2.21181218
```

Vamos a calcular ahora el standard error de los estimadores con la matriz de varianzas de los regresores:

```
Xa = cbind(d$mom_iq, d$mom_hs)
(Qa = 1/(n-1)*solve(var(Xa)))
```

```
##           [,1]           [,2]
## [1,] 1.115594e-05 -0.0001151616
## [2,] -1.151616e-04  0.0148740410
```

El standard error de los estimadores  $\hat{\beta}_1$  y  $\hat{\beta}_2$  son:

```
sqrt(diag(Qa)*sR2)
```

```
## [1] 0.06057408 2.21181218
```

Para  $\hat{\beta}_0$ :

```
( xmed = matrix(colMeans(Xa), ncol = 1) )
```

```
##           [,1]
## [1,] 100.0000000
## [2,]  0.7857143
```

```
sqrt( sR2*(1/n + 1/(n-1)*t(xmed) %*% solve(var(Xa)) %*% xmed ) )
```

```
##           [,1]
## [1,] 5.875208
```

Por último, R dispone de una función para calcular la matriz de varianzas de los parámetros estimados, es decir  $var(\hat{\beta}) = Q_{ii}\hat{s}_R^2$ , mediante:

```
vcov(m)
```

```
##           (Intercept)      mom_iq      mom_hs
## (Intercept) 34.51806922 -0.337161456 -0.05610582
## mom_iq      -0.33716146  0.003669219 -0.03787697
## mom_hs      -0.05610582 -0.037876974  4.89211314
```

Por tanto, el standard error de los estimadores será

```
sqrt(diag(vcov(m)))
```

```
## (Intercept)      mom_iq      mom_hs
## 5.87520802 0.06057408 2.21181218
```

Como vemos, los tres métodos dan el mismo resultado.

El valor de la t con  $n-k-1 = 431$  grados de libertad es

```
(t1 = qt(1-0.05/2, df = n-k-1))
```

```
## [1] 1.965483
```

El límite inferior (LI) y el límite superior de los intervalos será:

```
(LI = coef(m) - qt(1-0.05/2, df = n-k-1)*beta_se)
```

```
## (Intercept)      mom_iq      mom_hs  
## 14.1839148    0.4448487    1.6028370
```

```
(LS = coef(m) + qt(1-0.05/2, df = n-k-1)*beta_se)
```

```
## (Intercept)      mom_iq      mom_hs  
## 37.2791615    0.6829634   10.2973969
```

Si lo juntamos todo en una tabla

```
data.frame(estimacion = coef(m), se = beta_se, LI, LS)
```

```
##           estimacion      se      LI      LS  
## (Intercept) 25.731538 5.87520802 14.1839148 37.2791615  
## mom_iq      0.563906 0.06057408 0.4448487 0.6829634  
## mom_hs      5.950117 2.21181218 1.6028370 10.2973969
```

Directamente, mediante la función *confint()* de R se pueden obtener dichos valores:

```
confint(m)
```

```
##           2.5 %      97.5 %  
## (Intercept) 14.1839148 37.2791615  
## mom_iq      0.4448487 0.6829634  
## mom_hs      1.6028370 10.2973969
```

Si queremos otro nivel de confianza, por ejemplo, 90%:

```
confint(m, level = 0.90)
```

```
##           5 %      95 %  
## (Intercept) 16.0468646 35.4162117  
## mom_iq      0.4640559 0.6637562  
## mom_hs      2.3041730 9.5960608
```

- En el caso de la varianza del modelo. Su estimador es:

```
sR2
```

```
## [1] 328.9028
```

Y su intervalo de confianza:

```
c((n-k-1)*sR2/qchisq(1-0.05/2, df = n-k-1), (n-k-1)*sR2/qchisq(0.05/2, df = n-k-1))
```

```
## [1] 289.0557 377.6434
```