Taller 6: gran MCO

Econometría I

Paula Pereda (ppereda@correo.um.edu.uy)

18 de setiembre de 2020

`R` + MCO con un regresor

1. Tamaño de las clases & notas

Un investigador desea analizar la relación entre el tamaño de la clase (medido por la proporción de alumnos por profesor) y el puntaje promedio de la prueba. Por lo tanto, mide ambas variables en 10 clases diferentes y termina con los siguientes resultados.

- Tamaño de la clase: 23 19 30 22 23 29 35 36 33 25
- Puntaje de prueba: 430 430 333 410 390 377 325 310 328 375

Instrucciones

Cree los vectores cs (el tamaño de la clase) y ts (la puntuación de la prueba), que contengan las observaciones anteriores.

```
cs ← c(23, 19, 30, 22, 23, 29, 35, 36, 33, 25)
ts ← c(430, 430, 333, 410,    390, 377, 325, 310,    328, 375)
```

2. Estadísticos descriptivos

Media, varianza, covarianza y correlación

Los vectores cs y ts están disponibles en el ambiente (pueden comprobar esto: escriba el nombre de los objetos en la consola y presione enter).

Instrucciones

- Calcular la media, la varianza muestral y la desviación estándar muestral de ts.
- Calcular la covarianza y el coeficiente de correlación para ts y cs.

Sugerencia Utilice las funciones de R: mean(), sd(), cov(), cor() y var().

Media, varianza, covarianza y correlación

```r
# computa la media, varianza y desviación estándar de las notas
mean(ts)
```

```
#> [1] 370.8
```

```r
sd(ts)
```

```
#> [1] 44.75315
```

```r
var(ts)
```

```
#> [1] 2002.844
```

```r
# computa la covarianza y el coeficiente de correlación
cov(ts, cs)
```

```
#> [1] -251.4444
```

```r
cor(ts, cs)
```

```
#> [1] -0.9474424
```

3. Regresión lineal simple

Los vectores cs y ts están disponibles en el entorno de trabajo.

Instrucciones:

- Utilice lm() para estimar el modelo de regresión:

$$\text{TestScore}_i = \beta_0 + \beta_1 \text{ClassSize}_i + u_i$$

- Asigne el resultado a mod.
- Obtenga un resumen estadístico del modelo.

3. Regresión lineal simple

```r
# estime el modelo
# lm(ts ~ cs)

# le asigna el nombre mod

mod <- lm(ts ~ cs)

# obtiene un resumen del modelo
summary(mod)
```

```
#>
#> Call:
#> lm(formula = ts ~ cs)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -19.9248 -10.6002  -0.8506   5.8631  27.0246
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 567.4272    23.9606  23.682 1.08e-08 ***
#> cs           -7.1501     0.8536  -8.376 3.13e-05 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 15.19 on 8 degrees of freedom
#> Multiple R-squared:  0.8976,    Adjusted R-squared:  0.8849
#> F-statistic: 70.16 on 1 and 8 DF,  p-value: 3.132e-05
```

El modelo como objeto

Veamos cómo se estructura un objeto de clase lm.

Los vectores cs y ts, así como el modelo de objeto del ejercicio anterior, están disponibles en el ambiente.

Instrucciones:

- Use class() para chequear la clase del objeto mod.
- mod es un objeto de tipo lista con entradas con nombre. Verifique esto usando la función is.list().
- Vea qué información puede obtener de mod usando names().
- Leer una entrada arbitraria del objeto mod usando el operador $.

El modelo como objeto

```r
# chequee de qué clase es `mod`
class(mod)
```

```
#> [1] "lm"
```

```r
# use `is.list()` en `mod`
is.list(mod)
```

```
#> [1] TRUE
```

```r
# chequee qué contiene `mod` usando `names()`
names(mod)
```

```
#>  [1] "coefficients"  "residuals"     "effects"       "rank"
#>  [5] "fitted.values" "assign"        "qr"            "df.residual"
#>  [9] "xlevels"       "call"          "terms"         "model"
```

```r
# use el operador `$` en `mod`
## ejemplo:
mod$fitted.values
```

```
#>        1        2        3        4        5        6        7        8
#> 402.9754 431.5757 352.9248 410.1254 402.9754 360.0749 317.1744 310.0243
#>        9       10
#> 331.4746 388.6752
```

Graficando la regresión

Sabiendo que plot() se utiliza para graficar y permite ver la dispersión.

Instrucciones:

- Agregue la línea de regresión al diagrama de dispersión.
- El objeto mod está disponible en el ambiente.

Sugerencia Use la función abline().

Graficando la regresión

```
# agregue la línea de regresión al gráfico de puntos
plot(cs, ts)

abline(mod)
```

Resumen del modelo

Ahora lea y almacene parte de la información contenida en la salida de summary().

Instrucciones:

- Asigne la salida de resumen (mod) al objeto s.
- Compruebe los nombres de entrada de los objetos.
- Cree un nuevo objeto R2 y asigne el R2 de la regresión.
- El objeto mod está disponible en el ambiente.

Resumen del modelo

```r
# asigna el resumen del modelo al objeto `s`
s ← summary(mod)

# chequea el nombre de las entradas en `s`
names(s)
```

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

```r
# almacena el R^2 de ka regresión en el objeto `R2`
R2 ← s$r.squared
```

Coeficientes estimados

La función de resumen summary() también proporciona información sobre la significancia estadística de los coeficientes estimados.

<u>Instrucciones</u>:

- Extraiga la matriz de 2×4 con coeficientes estimados, errores estándar, estadísticos t y valores p correspondientes al resumen del modelo s
- Guarde esta matriz en un objeto llamado coefs
- Los objetos mod y s están disponibles en su ambiente.

```
# almacene los coeficientes de la matriz a `coefs`
coefs <- s$coefficients
```

8. Dejando el intercepto

Hasta ahora, hemos estimado modelos de regresión que consisten en un intercepto y un regresor único. En este ejercicio aprenderemos a especificar y estimar...

Tenga en cuenta que excluir el intercepto de un modelo de regresión puede ser una práctica poco fiable en algunas aplicaciones, ya que impone la función de expectativa condicional de t

Instrucciones:

- Averigüe cómo se debe especificar el argumento de la fórmula para una regresión de ts únicamente en cs, es decir, una regresión sin intersección. ¡Google es tu amigo!
- Estime el modelo de regresión sin intersección y almacene el resultado en mod_ni.
- Los vectores cs, ts y el modelo de objeto mod de ejercicios anteriores están disponibles en el ambiente.

8. Dejando el intercepto

```r
# regrese `ts` solamente en `cs`. Almacene el resultado en `mod_ni`.
mod_ni <- lm(ts ~ cs - 1)

# o
lm(ts ~ cs + 0)
```

```
#>
#> Call:
#> lm(formula = ts ~ cs + 0)
#>
#> Coefficients:
#>    cs
#> 12.65
```

9. El caso sin intercepto

En el ejercicio 8 ha estimado un modelo sin intersección. La función de regresión estimada es

$$\widehat{\mathbf{TestScore}} = \mathbf{12.65} \times \mathbf{ClassSize}$$

con un error estándar de 1.36.

Instrucciones:

- Convénzase de que todo es como se indicó anteriormente: extraiga la matriz de coeficientes del resumen de mod_ni y guárdelo en una variable llamada coef.
- Los vectores cs, ts y el objeto modelo mod_ni del ejercicio anterior están disponibles en su entorno de trabajo.

9. El caso sin intercepto

Sugerencia Se puede acceder a una entrada de una lista con nombre usando el operador $.

```
# extraiga la matriz de coeficientes del resumen del modelo y almacénelo como `coef`
coef ← summary(mod_ni)$coefficients

## ¡Correcto! Tenga en cuenta que solo hay una estimación de coeficiente (el coeficiente de cs)
## informado por resumen(mod_ni).
```

10. El caso sin intercepto

En los ejercicios 8 y 9 se ha tratado con un modelo sin intercepto. La función de regresión estimada fue

$$\widehat{\mathbf{TestScore}} = \mathbf{12.65} \times \mathbf{ClassSize}$$

con un error estándar de 1.36.

El coeficiente de la matriz de coeficientes del ejercicio 9 contiene el coeficiente estimado en ClassSize, su error estándar, el estadístico t de la prueba de significancia y el valor p correspondiente.

<u>Instrucciones</u>:

- Imprima el contenido de coef en la consola.
- Convénzase usted mismo de que el estadístico t informado es correcto: use las entradas de coef para calcular el estadístico t y guárdelo en t_stat.
- El coeficiente de matriz del ejercicio anterior está disponible en su ambiente.

10. El caso sin intercepto

Sugerencias

- Recuerden que $X[a,b]$ devuelve el elemento $[a,b]$ de la matriz X.
- El estadístico t para una prueba del tipo $H_0 : \beta_1 = 0$ se computa de la siguiente manera:

$$t = \frac{\hat{\beta}_1}{ee\left(\hat{\beta}_1\right)}$$

```r
# imprima los contenidos de `coef` en la consola
print(coef)
```

```
#>    Estimate Std. Error  t value     Pr(>|t|)
#> cs 12.65478    1.36013 9.304097 6.50056e-06
```

```r
# compute el estadístico t manualmente y asígnelo a `t_stat`
t_stat <- coef[1,1]/coef[1,2]
```

11. Dos regresiones, un gráfico

Los dos modelos de regresión estimados de los ejercicios anteriores son

$$\text{TestScore}_i = 12.65 \times \text{ClassSize}_i$$

y

$$\text{TestScore}_i = 567.4272 - 7.1501 \times \text{ClassSize}_i$$

Se le proporciona el código plot(cs, ts) que crea un diagrama de dispersión de ts y cs. Tenga en cuenta que esta línea debe ejecutarse antes que abline(), además, pueden colorear las líneas de regresión usando, por ejemplo, col = "red" o col = "blue" como un argumento adicional a abline() para una mejor distinción. Los vectores cs y ts, así como la lista de objetos mod y mod_ni de ejercicios anteriores, están disponibles en su ambiente.

Instrucciones:

- Genere un diagrama de dispersión de ts y cs y agregue las líneas de regresión estimadas de mod y mod_ni.

11. Dos regresiones, un gráfico

```r
# se grafica la línea de regresiòn de ambos modelos
plot(cs, ts)
abline(mod, col = "blue")
abline(mod_ni, col = "red")
```

12. SRy ST

Si la inspección gráfica no ayuda, los investigadores recurren a técnicas analíticas para detectar si un modelo se ajusta bien o mejor a los datos que otro modelo.

Volvamos al modelo de regresión simple que incluye un intercepto. La línea de regresión estimada para mod fue

$$\text{TestScore}_i = 567.43 - 7.15 \times \text{ClassSize}_i, R^2 = 0.8976, ESR = 15.19$$

Puede comprobar esto como mod y los vectores cs y ts están disponibles en su ambiente.

Instrucciones:

- Calcular la SR, la suma de los residuos al cuadrado y guárdelo en ssr.
- Calcular la ST, la suma total de cuadrados y guárdelo en tss.

Nota: `var ()` calcula la varianza de la muestra insesgada. => corrija esto multiplicando con (n-1) = 9

12. SR y ST

```
# compute la SR y almacénela en `ssr`
ssr ← sum(mod$residuals^2)

# compute la ST y almacénela en `tss`
tss ← 9*var(ts)
```

13. El R^2

El R^2 de la regresión guardada en mod es 0.8976. Puede verificar esto ejecutando summary(mod)$r.squared en la consola a continuación.

Recuerda que la fórmula de R^2 es:

$$R^2 = \frac{SE}{ST} = 1 - \frac{SR}{ST}$$

Instrucciones:

- Utilice ssr (SR) y tss (ST) para calcular R^2 a mano. Redondea el resultado a cuatro decimales y guárdelo en R2.
- Use el operador lógico == para verificar si su resultado coincide con el valor mencionado anteriormente.

13. El R^2

```r
# computa R^2, redonde a cuatro lugares después de la coma y guárdelo como "R2"
R2 <- round(1-ssr/tss,4)

# chequea si el resultado es correcto usando el operador "=="
R2 == 0.8976
```

```
#> [1] TRUE
```

14. El error estándar de una regresión

El error estándar de un modelo de regresión simple es

$$ESR = \frac{1}{n-2}\sum_{i=1}^{n}\widehat{u}_i^2 = \sqrt{\frac{SR}{n-2}}$$

El ESR mide el tamaño de un residuo promedio que es una estimación de la magnitud de un error de regresión típico.

El modelo de objeto mod y los vectores cs y ts están disponibles en su ambiente.

Instrucciones:

- Utilice summary() para obtener el ESR para la regresión de ts sobre cs guardado en el objeto modelo mod. Guarde el resultado en la variable SER.
- Utilice SER para calcular el SR y guárdelo en SSR.
- Compruebe que SSR es de hecho el SR comparando SSR con el resultado de sum(mod$residuals^2).

14. El error estándar de una regresión

```
# obtenga el SER usando `summary()` y guárdelo `SER`
SER <- summary(mod)$sigma

# compute el SSR y almacénelo como `SSR`
SSR <- SER^2*(length(ts)-2)

# haga la comapración
SSR == sum(mod$residuals^2)
```

```
#> [1] TRUE
```

15. La matriz de covarianza

Como se discute en el capítulo 4 del Stock & Watson, los estimadores MCO $\widehat{\boldsymbol{\beta}}_{\mathbf{0}}$ y $\widehat{\boldsymbol{\beta}}_{\mathbf{1}}$ son funciones del término de error aleatorio. Por tanto, son variables aleatorias en sí mismas. Para dos o más variables aleatorias, sus covarianzas y varianzas se resumen mediante una matriz de varianza-covarianza (que a menudo se denomina simplemente matriz de covarianza). Tomando la raíz cuadrada de los elementos diagonales de la matriz de covarianza estimada se obtiene $ee\left(\widehat{\beta}_0\right)$ y $ee\left(\widehat{\beta}_1\right)$, los errores estándar de $\widehat{\boldsymbol{\beta}}_{\mathbf{0}}$ y $\widehat{\boldsymbol{\beta}}_{\mathbf{1}}$.

summary() computa una estimación de esta matriz. La entrada respectiva en el output of summary (recuerden que summary() produce una lista) se llama cov.unscaled. El objeto del modelo mod is disponible en su ambiente. Instructions:

<u>Instrucciones</u>:

- Utilice summary() para obtener la estimación de la matriz de covarianzas para la regresión de los puntajes de las pruebas en las proporciones alumno-maestro almacenadas en el modelo de objeto mod. Guarde el resultado en cov_matrix.

- Obtenga los elementos diagonales de cov_matrix, calcule su raíz cuadrada y asigne el resultado al objeto SEs (errores estándar).

Sugerencia diag(A) devuelve un vector que contiene los elementos diagonales de la matriz A.

15. La matriz de covarianza

```
# obtenga la matriz y almacénela como `cov_matrix`
cov_matrix ← summary(mod)$cov.unscaled

# compute los errores estándar y asignelos en el vector `SEs`
SEs ← sqrt(diag(cov_matrix))
```

16. Testeando hipótesis nulas

Considere el modelo de regresión estimado:

$$\text{TestScore}_i = 567.43 - 7.15 \times \text{ClassSize}_i, R^2 = 0.8976, ESR = 15.19$$

con $ee\left(\widehat{\beta}_0\right) = 23.96$ y $ee\left(\widehat{\beta}_1\right) = 0.85$

Instrucciones:

- Compute el valor-p para una prueba t de la hipótesis de que el intercepto es cero frente a la alternativa de dos lados que no es cero. Guarde el resultado en p_int.
- Compute el valor-p para una prueba t de la hipótesis de que ClassSize es cero frente a la alternativa de dos lados que no es cero. Guarde el resultado en p_STR.

Sugerencia Ambas hipótesis se pueden probar individualmente mediante una prueba de dos lados. Utilice pnorm() para obtener probabilidades acumuladas de resultados estándar distribuidos normalmente.

16. Testeando hipótesis nulas

```
# compute el valor-p para el primer test de significancia y guárdelo en p_int
t_int ← 567.43/23.9606
p_int ← 2*(1-pnorm(abs(t_int)))

# compute el valor-p para el segundo test de significancia y guárdelo en p_cs
t_STR ← 7.15/0.8536
p_STR ← 2*(1-pnorm(abs(t_STR)))
```

17. Testeando hipótesis nulas

Considere de nuevo el modelo de regresión estimado:

$$\text{TestScore}_i = 567.43 - 7.15 \times \text{ClassSize}_i, R^2 = 0.8976, ESR = 15.19$$

con $ee\left(\widehat{\beta}_0\right) = 23.96$ y $ee\left(\widehat{\beta}_1\right) = 0.85$

¿Se puede rechazar la hipótesis nula discutida en el ejercicio anterior usando pruebas t individuales al 5%? Los objetos t_int y t_STR son los estadísticos t. Ambos están disponibles en el ambiente.

<u>Instrucciones</u>: Junte t_int y t_STR en un vector y use operadores lógicos para verificar si se aplica la regla de rechazo correspondiente.

Sugerencia Ambas pruebas son pruebas t bilaterales. El concepto clave 5.2 resume cómo se realiza una prueba t bilateral. Utilice qnorm() para obtener valores críticos normales estándar.

17. Testeando hipótesis nulas

```r
test <- c(t_int, t_STR)
# el resultado es `TRUE` si se rechaza la hipótesis nula
abs(test) >= qnorm(0.975)
```

```
#> [1] TRUE TRUE
```

18. Intervalo de confianza

mod, el objeto de clase lm contiene los resultados de la siguiente regresión

$$\text{TestScore }_i = 567.43 - 7.15 \times \text{ ClassSize }_i, R^2 = 0.8976, ESR = 15.19$$

con $ee\left(\widehat{\beta}_0\right) = 23.96$ y $ee\left(\widehat{\beta}_1\right) = 0.85$

está en su ambiente de trabajo

<u>Instrucciones</u>: Compute intervalos de confianza al 90% para ambos coeficientes.

Sugerencia Use la función confint(), vea ?confint. El argumento level establece el nivel de confianza que se utilizará.

```
confint(mod, level = 0.9)
```

```
#>                     5 %       95 %
#> (Intercept) 522.87120 611.983142
#> cs           -8.73742  -5.562738
```

`R` + MCO con múltiples regresores

1. Regresión lineal múltiple

En el transcurso de esta sección, trabajará con Boston, el conjunto de datos de Boston Housing que contiene 506 observaciones sobre el valor de las viviendas en los suburbios de Boston. Boston viene con el paquete MASS.

Instrucciones:

- Cargue tanto el paquete como el conjunto de datos.
- Obtenga una descripción general de los datos utilizando funciones conocidas de los prácticos anteriores.
- Estime un modelo de regresión lineal simple que explique el valor medio de la vivienda de los distritos (medv) por los porcentajes de hogares con un nivel socioeconómico bajo, lstat y una constante. Guarde el modelo en bh_mod.
- Imprima un resumen de coeficientes en la consola que informa de errores estándar robustos.

Sugerencia Solo necesitarán funciones básicas de R: library(), data(), lm() y coeftest().

1. Regresión lineal múltiple

```r
# cargue el paquete y los datos
library(MASS)
data("Boston")
# chequeen los datos
summary(Boston)
```

```
#>       crim                zn             indus            chas        
#>  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
#>  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
#>  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
#>  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
#>  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
#>  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
#>       nox               rm             age              dis        
#>  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
#>  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
#>  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
#>  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
#>  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
#>  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
#>       rad              tax           ptratio          black       
#>  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
#>  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
#>  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
#>  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
#>  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
#>  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
#>      lstat            medv      
#>  Min.   : 1.73   Min.   : 5.00  
#>  1st Qu.: 6.95   1st Qu.:17.02  
#>  Median :11.36   Median :21.20  
```

1. Regresión lineal múltiple

```r
# o
str(Boston)
```

```
#> 'data.frame':	506 obs. of  14 variables:
#>  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
#>  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
#>  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
#>  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
#>  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
#>  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
#>  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
#>  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
#>  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
#>  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
#>  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
#>  $ black  : num  397 397 393 395 397 ...
#>  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
#>  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```r
# o
head(Boston)
```

```
#>      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
#> 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
#> 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
#> 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
#> 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
#> 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
#> 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
#>   medv
```

1. Regresión lineal múltiple

```r
# estime un modelo de regresión simple
bh_mod <- lm(medv ~ lstat, data = Boston)
# imprima en la consola el resumen
coeftest(bh_mod, vcov. = vcovHC)
```

```
#>
#> t test of coefficients:
#>
#>              Estimate Std. Error t value  Pr(>|t|)
#> (Intercept) 34.55384    0.75857  45.552 < 2.2e-16 ***
#> lstat       -0.95005    0.05008 -18.971 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
R2_res <- summary(bh_mod)$r.squared
```

2. Regresión lineal múltiple

Ahora, ampliemos el enfoque del ejercicio anterior agregando regresores adicionales al modelo y estimándolo nuevamente.

Como se discute en el capítulo 6, agregar regresores al modelo mejora el ajuste de modo que el ESR disminuye y el R^2 aumenta. El objeto modelo bh_mod está disponible en el ambiente.

Instrucciones:

- Regresar el valor medio de la vivienda en un distrito, medv, sobre la edad promedio de los edificios, la edad, la tasa de delincuencia per cápita, crim, el porcentaje de personas con un nivel socioeconómico bajo, lstat y una constante. Dicho de otra manera, estima el modelo

$$\mathbf{med}\, v_i = \beta_0 + \beta_1 \mathbf{lstat}_i + \beta_2\ \mathbf{age}\ _i + \beta_3\ \mathbf{crim}_i + u_i$$

- Imprima un resumen de coeficientes en la consola que informa errores estándar robustos para el nuevo modelo.

- El R^2 del modelo de regresión simple se almacena en R2_res. Guarde el R^2 del modelo de regresión múltiple en R2_unres y compruebe si el modelo de regresión múltiple produce un mayor R^2.

2. Regresión lineal múltiple

```r
# estime la regresión
mod <- lm(medv ~ lstat + crim + age, data = Boston)

# obtenga un resumen de los coeficientes estimados
coeftest(mod, vcov. = vcovHC)
```

```
#> 
#> t test of coefficients:
#> 
#>              Estimate Std. Error  t value Pr(>|t|)    
#> (Intercept) 32.828045   0.751505  43.6831  < 2e-16 ***
#> lstat       -0.994091   0.083058 -11.9686  < 2e-16 ***
#> crim        -0.082622   0.029733  -2.7788  0.00566 ** 
#> age          0.037647   0.016930   2.2236  0.02662 *  
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# compare los R2
R2_unres <- summary(mod)$r.squared
R2_unres < R2_res
```

```
#> [1] FALSE
```

3. Test de hipótesis

Reconsidere el modelo estimado en el ejercicio anterior

$$\widehat{medv}_i = 32.828 - 0.994 \times lstat_i - 0.083 \times crim_i + 0.038 \times age_i$$

con $ee\left(\widehat{\beta}_0\right) = 0.75$, $ee\left(\widehat{\beta}_1\right) = 0.05$, $ee\left(\widehat{\beta}_3\right) = 0.04$ y $ee\left(\widehat{\beta}_4\right) = 0.01$

Al igual que en el marco de regresión lineal simple, podemos realizar pruebas de hipótesis sobre los coeficientes en modelos de regresión múltiple. La hipótesis más común es $H_0 : \beta_j = 0$ contra la alternativa $H_1 : \beta_j \neq 0$ para algunos j en $0, 1, \ldots, k$.

Las estimaciones de los coeficientes, así como los errores estándar correspondientes, están disponibles en coefs y SE, respectivamente.

Instrucciones:

- Calcule el estadístico t para cada coeficiente utilizando los objetos predefinidos coefs y SEs. Asígnelos a tstats.
- Calcule los valores-p para cada coeficiente y asígnelos a pval.
- Compruebe con la ayuda de operadores lógicos si las hipótesis se rechazan al 1% de significancia.

3. Test de hipótesis

Sugerencias

- El estadístico t para cada coeficiente se define como $t = \frac{\widehat{\beta}_j - \beta_{j,0}}{SE\left(\hat{\beta}_j\right)}$
- El valor-p para un test bilateral se computa como $2 \cdot \Phi\left(-\left|t^{act}\right|\right)$ donde t^{act} denota el estadístico t computado.

```
coefs <- mod$coefficients
SEs <- coef(summary(mod))[, "Std. Error"]

# compute el estadístico t para cada coeficiente, asígnelos a `tstat`
tstats <- coefs/SEs

# compute los valores-p para todos los test de significancia, asígnelos a `pval`
pvals <- 2*(pnorm(-abs(tstats)))

# comprobar si las hipótesis se rechazan al 1% de nivel de significancia
pvals < 0.01
```

```
#> (Intercept)       lstat        crim         age
#>        TRUE        TRUE       FALSE        TRUE
```

4. Intervalos de confianza

Reconsidere el modelo estimado en el ejercicio anterior

$$\widehat{medv_i} = 32.828 - 0.994 \times lstat_i - 0.083 \times crim_i + 0.038 \times age_i$$

con $ee\left(\widehat{\beta}_0\right) = 0.75$, $ee\left(\widehat{\beta}_1\right) = 0.05$, $ee\left(\widehat{\beta}_3\right) = 0.04$ y $ee\left(\widehat{\beta}_4\right) = 0.01$

que está disponible en el ambiente como mod.

Instrucciones:

Construya intervalos de confianza al 99% para todos los coeficientes del modelo. Use los intervalos para decidir si las hipótyesis nulas $H_0 : \beta_j = 0, j = 0, 1, 2, 3, 4$ se rechazan al nivel de 1%.

Sugerencia Use la función confint(), vea ?confint. El argumento level establece el nivel de confianza que se utilizará.

```
confint(mod, level = 0.99)
```

```
#>                    0.5 %      99.5 %
#> (Intercept) 30.894648500 34.76144090
#> lstat       -1.125316516 -0.86286535
#> crim        -0.175559754  0.01031531
#> age          0.005976671  0.06931660
```

5. Hipótesis conjuntas

A veces nos interesa probar hipótesis conjuntas que imponen restricciones sobre coeficientes de las regresiones múltiples. Por ejemplo, en el modelo

$$\mathbf{med}\, v_i = \beta_0 + \beta_1 \times \mathbf{lstat}_i + \beta_2 \times \mathbf{crim}_i + \beta_3 \times \boldsymbol{age}_i + \boldsymbol{u}_i$$

podemos testear $H_0 : \beta_2 = \beta_3$ versus la alternativa $H_1 : \beta_2 \neq \beta_3$ (que es una hipótesis conjunta ya que imponemos una restricción a dos coeficientes de regresión).

La idea básica detrás de probar tal hipótesis es realizar dos regresiones y comparar los resultados: para una de las regresiones, imponemos las restricciones de formalizado por la nula (llamamos a este modelo de regresión restringido), mientras que para la otra regresión la restricción queda fuera (a esto lo llamamos el modelo sin restricciones). A partir de este punto de partida construimos un estadístico de prueba que, bajo la nula, sigue una distribución bien conocida, una distribución F.

Sin embargo, en este ejercicio comenzamos con los cálculos iniciales necesarios para construir la prueba estadística.

Instrucciones:

- Estimar el modelo restringido, es decir, el modelo donde la restricción formalizada por $H_0 : \beta_2 = \beta_3$ se asume que es cierto. Guarde el modelo en model_res.

5. Hipótesis conjuntas

Instrucciones:

- Compute el SR del modelo restringido y asígnelo a RSSR.
- Estime el modelo sin restringir es decir, el modelo donde se asume que la restricción es falsa. Guárdelo en model_unres.
- Compute el SR del modelo sin restringir y asígnelo a USSR.

Sugerencias

- El modelo restringido puede estar escrito como

$$\mathbf{medv}_i = \beta_0 + \beta_1 \times \ \mathbf{lstat}\ _i + \beta_2 \times \mathbf{crim}_i + \beta_2 \times \boldsymbol{age}_i + \boldsymbol{u}_i$$

que, después de reorganizar, se puede expresar como

$$\mathbf{medv}_i = \beta_0 + \beta_1 \times \ \mathbf{lstat}_i + \beta_2 \times (\mathbf{crim}_i + \boldsymbol{age}_i) + \boldsymbol{u}_i$$

- Tenga en cuenta que los residuos de un modelo de regresión están disponibles como residuos en el objeto lm correspondiente. Entonces puede acceder a ellos como de costumbre a través del operador $.

5. Hipótesis conjuntas

```r
# estime el modelo restringido y guárdelo como `model_res`
model_res ← lm(medv ~ lstat + I(crim + age), data = Boston)

# compute el SR del modelo restringido y asígnelo a RSSR
RSSR ← sum(model_res$residuals^2)

# estime el modelo irrestricto y guárdelo como `model_unres`
model_unres ← lm(medv ~ lstat + crim + age, data = Boston)

# compute el SR del modelo irrestricto y asígnelo a `USSR`
USSR ← sum(model_unres$residuals^2)
```

5. Hipótesis conjuntas

Después de estimar los modelos y calcular los SSR, ahora tiene que calcular el estadística de prueba y realizar la prueba F. Como se mencionó en el último ejercicio, el estadístico de prueba sigue una distribución F. Más precisamente, nos ocupamos de una distribución $F_{q,n-k-1}$ donde q denota el número de restricciones bajo hipótesis nula y k es número de regresores en el modelo sin restricciones, excluyendo el intercepto.

Instrucciones:

- Compute el estadístico F y asígnelo como Fstat.
- Compute el valor-p y asígnelo como apval.
- Chequee si la hipotesis nula si rechaza a un nivel de 1% usando operadores lógicos.
- Verifique sus resultados usando linearHypothesis() e imprimiendo los resultados.

5. Hipótesis conjuntas

```r
# compute el estadístico y asígnelo a `Fstat`
Fstat <- ((RSSR-USSR)/1)/(USSR/(nrow(Boston)-3-1))

# compute el valor-p y asígnelo a `pval`
pval <- 1 - pf(Fstat, df1 = 1, df2 = nrow(Boston)-3-1)

# chequee si la hipotesis nula si rechaza a un nivel de 1% usando operadores lógicos.
pval < 0.01
```

```
#> [1] TRUE
```

```r
# verifique ese resultado con `linearHypothesis()`
linearHypothesis(model_unres, "age = crim")
```

```
#> Linear hypothesis test
#>
#> Hypothesis:
#> - crim  + age = 0
#>
#> Model 1: restricted model
#> Model 2: medv ~ lstat + crim + age
#>
#>   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
#> 1    503 19324
#> 2    502 18968  1    355.14 9.3989 0.002288 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```