3. Greene, William, H. *Econometric Analysis, Seventh Edition.* Prentice Hall, 2012. Application 2, Page 106. Datos: pages.stern.nyu.edu/~wgreene/Text/Edition7/TableF4-4.txt.

Christensen and Greene (1976) estimaron una función de costo Cobb-Douglas generalizada para la generación de electricidad de la forma

$$\ln C = \alpha + \beta \ln Q + \gamma \left[\frac{1}{2}(\ln Q)^2\right] + \delta_k \ln P_k + \delta_l \ln P_l + \delta_f \ln P_f + \varepsilon$$

P_k, P_l y P_f son los precios de una unidad de capital, trabajo, y combustible, respecitvamente. Q es la cantidad producida y C el costo total.

$$b = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta_k \\ \delta_l \\ \delta_f \end{pmatrix}$$

Para cumplir con la teoría de producción empleada, es necesario imponer la restricción de que la función de costo es homogenea de grado uno en los precios:

$$\delta_k + \delta_l + \delta_f = 1$$

```
datos <- read.table("https://pages.stern.nyu.edu/~wgreene/Text/Edition7/TableF4-4.txt",
                    skip = 1,
                    header = T)
stargazer(datos, header = F,
          title = "Datos en Christensen and Greene (1976)")
```

Table 1: Datos en Christensen and Greene (1976)

Statistic	N	Mean	St. Dev.	Min	Max
id	158	127.209	60.792	1	218
year	158	1,970.000	0.000	1,970	1,970
cost	158	53.270	87.059	0.130	737.409
q	158	10,469.410	15,187.520	4.000	115,500.000
pl	158	8,001.863	1,398.837	5,063.490	13,044.000
sl	158	0.139	0.055	0.046	0.329
pk	158	71.421	11.977	31.725	92.650
sk	158	0.226	0.061	0.092	0.457
pf	158	30.746	7.943	9.000	51.463
sf	158	0.632	0.083	0.244	0.814

i) Obtener el estimador de MCO, $\hat{b}_{MCU}$ ignorando la restriccion.

```
# variables a las que aplicaremos logaritmo natural
my_variables <- c("cost", "q", "pk", "pl", "pf")

# aplicamos logaritmos y unimos a la base de datos
logs <- sapply(datos[my_variables], log)
colnames(logs) <- paste("ln", my_variables, sep = "_")
```

```r
datos <- cbind(datos, logs)
datos$ln_q.sq <- 1/2*datos$ln_q^2

# estimamos mínimos cuadrados
u_model <- lm(ln_cost ~ ln_q + ln_q.sq + ln_pk + ln_pl + ln_pf,
              data = datos)

parametros <- c("$\\hat{\\alpha}_{MCU}$", "$\\hat{\\beta}_{MCU}$",
                "$\\hat{\\gamma}_{MCU}$","$\\hat{\\delta_k}_{MCU}$",
                "$\\hat{\\delta_l}_{MCU}$", "$\\hat{\\delta_f}_{MCU}$")

stargazer(u_model, header = F, title = "Modelo no restringido", no.space = TRUE,
          covariate.labels = c(parametros[-1], parametros[1]),
          dep.var.labels = "$\\ln{C}$")
```

Table 2: Modelo no restringido

	Dependent variable:
	$\ln C$
$\hat{\beta}_{MCU}$	0.403^{***}
	(0.032)
$\hat{\gamma}_{MCU}$	0.061^{***}
	(0.004)
$\hat{\delta_k}_{MCU}$	0.157^{***}
	(0.058)
$\hat{\delta_l}_{MCU}$	0.146^{**}
	(0.070)
$\hat{\delta_f}_{MCU}$	0.685^{***}
	(0.043)
$\hat{\alpha}_{MCU}$	-6.739^{***}
	(0.706)
Observations	158
R^2	0.992
Adjusted R^2	0.992
Residual Std. Error	0.138 (df = 152)
F Statistic	$3{,}880.407^{***}$ (df = 5; 152)
Note:	*p<0.1; **p<0.05; ***p<0.01

ii) Obtener el estimador de MCO restringido $\hat{b}_{MCR}$ con los comandos del paquete estadístico que se esté utilizando.

Podemos utilizar la librería de nombre `pracma`; requerimos especificar la restricción en la forma $Rb = r$. Sabemos que: $R = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}_{1\times 6}$ y $r = 1$. El código para generar los estimadores restringidos es:

```r
# R
R <- matrix(data = c(0,0,0,1,1,1), nrow = 1)

# r
r <- 1
```

```r
regresoras <- c("ln_q", "ln_q.sq", "ln_pk", "ln_pl", "ln_pf")

# Construimos la matrix X de dim nxk
X <- cbind(1, datos[regresoras])
names(X)[1] <- "1s"
X <- as.matrix(X)

#estimamos
coefs_restringidos <- pracma::lsqlincon(X, datos$ln_cost, Aeq =  R, beq =  r)

matrix(coefs_restringidos,
       dimnames = list(c("intercepto", regresoras), "estimadores"))
```

```
##            estimadores
## intercepto -6.81816332
## ln_q        0.40274543
## ln_q.sq     0.06089514
## ln_pk       0.16203385
## ln_pl       0.15244470
## ln_pf       0.68552145
```

iii) Obtener el estimador de MCO restringido $\hat{b}_{MCR}$ con la fórmula:

$$\hat{b}_{MCR} = \hat{b}_{MCU} - (X'X)^{-1}R'\left[R(X'X)^{-1}R'\right]^{-1}(R\hat{b}_{MCU} - r)$$

El siguiente código calcula los estimadores restringidos:

```r
library(xtable)

#b_mcu
b_mcu <- as.matrix(u_model$coefficients)

# estimamos b_mcr
b_mcr <- b_mcu - solve(t(X)%*%X)%*%t(R)%*%solve(R%*%solve(t(X)%*%X)%*%t(R))%*%(R%*%b_mcu-r)

# presentamos en una tabla
rownames(b_mcr) <- gsub("MCU", "MCR", parametros)
b_mcr <- as.data.frame(b_mcr)
names(b_mcr) <- "Valor"
print(xtable(b_mcr, digits = 8),
      sanitize.text.function=function(x){x},
      comment = F)
```

	Valor
$\hat{\alpha}_{MCR}$	-6.81816332
$\hat{\beta}_{MCR}$	0.40274543
$\hat{\gamma}_{MCR}$	0.06089514
$\hat{\delta_k}_{MCR}$	0.16203385
$\hat{\delta_l}_{MCR}$	0.15244470
$\hat{\delta_f}_{MCR}$	0.68552145

iv) Verificar con los datos obtenidos:

a) $\hat{U}_{MCR} = \hat{U}_{MCU} - X(\hat{b}_{MCR} - \hat{b}_{MCU})$

Reordenemos y probemos lo siguiente: $\hat{U}_{MCR} - \left(\hat{U}_{MCU} - X(\hat{b}_{MCR} - \hat{b}_{MCU})\right) = 0$, recordando que

$$\hat{U}_{MCR} \equiv Y - X\hat{b}_{MCR}$$

```
b_mcr <- as.matrix(b_mcr)
U_mcr <- as.matrix(datos$ln_cost - X%*%b_mcr)
U_mcu <- as.matrix(resid(u_model))

#lado derecho de la igualdad
B <- U_mcu - X%*%(b_mcr - b_mcu)

head(round(U_mcr - B, 10))
```

```
##      Valor
## [1,]     0
## [2,]     0
## [3,]     0
## [4,]     0
## [5,]     0
## [6,]     0
```

```
tail(round(U_mcr - B, 10))
```

```
##        Valor
## [153,]     0
## [154,]     0
## [155,]     0
## [156,]     0
## [157,]     0
## [158,]     0
```

```
all(U_mcr - B<0.00000000000001)
```

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

b) $\hat{U}'_{MCR}\hat{U}_{MCR} - \hat{U}'_{MCU}\hat{U}_{MCU} = (\hat{b}_{MCR} - \hat{b}_{MCU})'X'X(\hat{b}_{MCR} - \hat{b}_{MCU})$.

```
# lado izquierdo de la igualdad
A <- t(U_mcr)%*%U_mcr - t(U_mcu)%*%U_mcu

# lado derecho de la igualdad
B <- t(b_mcr - b_mcu)%*%t(X)%*%X%*%(b_mcr - b_mcu)

# comprobamos que A = B
abs(A - B) < 0.0000000000001
```

```
##       Valor
## Valor  TRUE
```

c) $\hat{U}'_{MCR}\hat{U}_{MCR} - \hat{U}'_{MCU}\hat{U}_{MCU} = (R\hat{b}_{MCU} - r)'\left[R(X'X)^{-1}R'\right]^{-1}(R\hat{b}_{MCU} - r)$

```
# lado derecho de la igualdad
B <- t(R%*%b_mcu - r)%*%solve(R%*%solve(t(X)%*%X)%*%t(R))%*%(R%*%b_mcu-r)

# comprobamos que A = B (se calculó A en inciso anterior)
abs(A - B) < 0.0000000000001
```

```
##       Valor
## Valor  TRUE
```

v) Obtener el estimador de MCO restringido $\hat{b}_{MCR}$ insertando la restricción $\delta_f = 1 - \delta_k - \delta_l$ en la ecuación original para estimar por MCO la regresión:

$$\ln \frac{C}{P_f} = \alpha + \beta \ln Q + \gamma \left[\frac{1}{2}(\ln Q)^2\right] + \delta_k \ln \frac{P_k}{P_f} + \delta_l \ln \frac{P_l}{P_f} + \varepsilon$$

Desarrollamos:

$$\begin{aligned}
\ln C &= \alpha + \beta \ln Q + \gamma \left[\frac{1}{2}(\ln Q)^2\right] + \delta_k \ln P_k + \delta_l \ln P_l + \underline{\delta_f} \ln P_f + \varepsilon \\
&= \alpha + \beta \ln Q + \gamma \left[\frac{1}{2}(\ln Q)^2\right] + \delta_k \ln P_k + \delta_l \ln P_l + \underline{(1 - \delta_k - \delta_l)} \ln P_f + \varepsilon \\
&= \alpha + \beta \ln Q + \gamma \left[\frac{1}{2}(\ln Q)^2\right] + \delta_k \ln P_k + \delta_l \ln P_l + \underline{\ln P_f - \delta_k \ln P_f - \delta_l \ln P_f} + \varepsilon
\end{aligned}$$

Reagrupamos términos:

$$\begin{aligned}
\ln C &= \alpha + \beta \ln Q + \gamma \left[\frac{1}{2}(\ln Q)^2\right] + \delta_k(\ln P_k - \ln P_f) + \delta_l(\ln P_l - \ln P_f) + \ln P_f + \varepsilon \\
&= \alpha + \beta \ln Q + \gamma \left[\frac{1}{2}(\ln Q)^2\right] + \delta_k \ln \frac{P_k}{P_f} + \delta_l \ln \frac{P_l}{P_f} + \ln P_f + \varepsilon \\
\implies \ln C - \ln P_f &= \alpha + \beta \ln Q + \gamma \left[\frac{1}{2}(\ln Q)^2\right] + \delta_k \ln \frac{P_k}{P_f} + \delta_l \ln \frac{P_l}{P_f} + \varepsilon \\
\therefore \ln \frac{C}{P_f} &= \alpha + \beta \ln Q + \gamma \left[\frac{1}{2}(\ln Q)^2\right] + \delta_k \ln \frac{P_k}{P_f} + \delta_l \ln \frac{P_l}{P_f} + \varepsilon
\end{aligned}$$

Estimamos por MCO:

(siguiente página)

```
r_model_MCO <- lm(I(log(cost/pf)) ~ ln_q + ln_q.sq + I(log(pk/pf)) + I(log(pl/pf)),
                  data = datos)

stargazer(r_model_MCO, header = F,
          covariate.labels = c(gsub("MCU", "MCR", parametros)[c(-1,-6)],
                               gsub("MCU", "MCR", parametros)[1]),
          dep.var.labels = "$\\ln{C/P_F}$",
          title = "Modelo con restricción estimado por MCO",
          no.space = T
          )
```

Table 3: Modelo con restricción estimado por MCO

	Dependent variable:
	$\ln C/P_F$
$\hat{\beta}_{MCR}$	0.403^{***}
	(0.031)
$\hat{\gamma}_{MCR}$	0.061^{***}
	(0.004)
$\hat{\delta_k}_{MCR}$	0.162^{***}
	(0.040)
$\hat{\delta_l}_{MCR}$	0.152^{***}
	(0.047)
$\hat{\alpha}_{MCR}$	-6.818^{***}
	(0.252)
Observations	158
R^2	0.992
Adjusted R^2	0.992
Residual Std. Error	0.138 (df = 153)
F Statistic	$4{,}879.590^{***}$ (df = 4; 153)
Note:	*p<0.1; **p<0.05; ***p<0.01

ii) Generar (usando algún paquete econométrico) una muestra i.i.d. $\{X_i\}_{i=1}^{10} \sim N(\mu, 0.04)$, donde $\mu = -1.2$

```
n <- 10
mu <- -1.2
sigma <- 0.04
muestra <- rnorm(n = n, mean = mu, sd = sqrt(sigma))
```

iii) Con los datos $\{X_i\}_{i=1}^{10}$ obtenidos en ii), encontrar el intervalo de confianza de μ, obtenido en i), con nivel de confianza del 90%. ¿Se encuentra -1.2 dentro de este intervalo?

Necesitamos unos q_1 y q_2 tales que la probabilidad sea del 90%, es decir, que el área bajo la curva sea de 0.9.

```
# 6% de la probabilidad
alpha1 <- 0.06
q1 <- qnorm(alpha1)

# 4% de la probabilidad
alpha2 <- .1 - alpha1
q2 <- -qnorm(alpha2)

c(q1, q2)
```

```
## [1] -1.554774  1.750686
```

Por tanto, dado que el intervalo de confianza calculado en i) es $\left(\overline{X}_{10} - \sqrt{\frac{1}{250}}q_2,\ \overline{X}_{10} - \sqrt{\frac{1}{250}}q_1\right)$, el intervalo con los valores elegidos de q_1 y q_2 sería:

```
CI <- c(mean(muestra) - sqrt(1/250)*q2, mean(muestra) - sqrt(1/250)*q1)
CI
```

```
## [1] -1.284283 -1.075227
```

Es decir:

$$CI = \left(-1.2842826,\ -1.0752269\right)$$

Entonces, es obvio que $-1.2 \in CI$

iv) Repetir 999 veces los pasos ii) y iii) para obtener en total 1000 intervalos de confianza de μ con nivel de confianza del 90%. ¿Qué porcentaje de estos 1000 intervalos incluye a -1.2?

```
intervalos <- data.frame(t(CI))

for (i in 1:999) {

  muestra <- rnorm(n = n, mean = mu, sd = sqrt(sigma))
  ci <- c(mean(muestra) - sqrt(1/250)*q2, mean(muestra) - sqrt(1/250)*q1)
  intervalos <- rbind(intervalos, ci)

}

porcentaje <- mean(intervalos[1] < (-1.2) & (-1.2) < intervalos[2])
porcentaje
```

```
## [1] 0.89
```

Es decir, el porcentaje de los 1000 intervalos que contiene a -1.2 es 89%

v) Usando el mismo pivote Q en i), encontrar otro (i.e. usar otra selección de q_1 y q_2 en $\mathbb{P}[q_1 < Q(X_1, X_2, \ldots, X_{10}; \mu) < q_2] = 0.90$) intervalo de confianza de $\big(T_1(X_1, X_2, \ldots, X_n), T_2(X_1, X_2, \ldots, X_n)\big)$ con 90% de nivel de confianza.

```
# 5% de la probabilidad
alpha1 <- 0.05
q1.2 <- qnorm(alpha1)

# 5% de la probabilidad
alpha2 <- .1 - alpha1
q2.2 <- -qnorm(alpha2)

c(q1.2, q2.2)
```

```
## [1] -1.644854  1.644854
```

Por tanto, dado que el intervalo de confianza calculado en i) es $\big(\overline{X}_{10} - \sqrt{\frac{1}{250}}q_2,\ \overline{X}_{10} - \sqrt{\frac{1}{250}}q_1\big)$, el intervalo con los valores elegidos de $q_{1.2}$ y $q_{2.2}$ sería:

```
CI.2 <- c(mean(muestra) - sqrt(1/250)*q2.2, mean(muestra) - sqrt(1/250)*q1.2)
CI.2
```

```
## [1] -1.228582 -1.020523
```

Es decir:

$$CI.2 = \big(-1.2285819,\ -1.0205225\big)$$

ii) Generar (usando algún paquete econométrico) una muestra i.i.d. $\{X_i\}_{i=1}^{10} \sim N(1, \sigma^2)$, donde $\sigma^2 = 0.04$

```
n <- 10
mu <- 1
sigma <- 0.04
muestra <- rnorm(n = n, mean = mu, sd = sqrt(sigma))
```

iii) Con los datos $\{X_i\}_{i=1}^{10}$ obtenidos en ii), encontrar el intervalo de confianza de σ^2, obtenido en i), con nivel de confianza del 90%. ¿Se encuentra 0.04 dentro de este intervalo?

Necesitamos unos q_1 y q_2 tales que la probabilidad sea del 90%, es decir, que el área bajo la curva sea de 0.9.

```
# 6% de la probabilidad
alpha1 <- 0.06
q1 <- qchisq(p = 1-alpha1, df = n, lower.tail = F)

# 4% de la probabilidad
alpha2 <- .1 - alpha1
q2 <- qchisq(p = 1-alpha2, df = n)

c(q1, q2)
```

```
## [1]  4.156724 19.020743
```

Por tanto, dado que el intervalo de confianza calculado en i) es $\left(\frac{\sum_{i=1}^{10}(X_i-1)^2}{q_2}, \ \frac{\sum_{i=1}^{10}(X_i-1)^2}{q_1}\right.$, el intervalo con los valores elegidos de q_1 y q_2 sería:

```
CI <- c(sum((muestra-1)^2)/q2, sum((muestra-1)^2)/q1)
CI
```

```
## [1] 0.01415011 0.06474947
```

Es decir:

$$CI = \left(0.0141501, \ 0.0647495\right)$$

Entonces, es obvio que $0.04 \in CI$

iv) Repetir 999 veces los pasos ii) y iii) para obtener en total 1000 intervalos de confianza de μ con nivel de confianza del 90%. ¿Qué porcentaje de estos 1000 intervalos incluye a -1.2?

```
intervalos <- data.frame(t(CI))

for (i in 1:999) {

  muestra <- rnorm(n = n, mean = mu, sd = sqrt(sigma))
  ci <- c(sum((muestra-1)^2)/q2, sum((muestra-1)^2)/q1)
  intervalos <- rbind(intervalos, ci)

}

porcentaje <- mean(intervalos[1] < (0.04) & (0.04) < intervalos[2])
porcentaje
```

```
## [1] 0.905
```

Es decir, el porcentaje de los 1000 intervalos que contiene a 0.04 es 90.5%

v) Usando el mismo pivote Q en i), encontrar otro (i.e. usar otra selección de q_1 y q_2 en $\mathbb{P}[q_1 < Q(X_1, X_2, \ldots, X_{10}; \mu) < q_2] = 0.90$) intervalo de confianza de $\big(T_1(X_1, X_2, \ldots, X_n), T_2(X_1, X_2, \ldots, X_n)\big)$ con 90% de nivel de confianza.

```
# 5% de la probabilidad
alpha1 <- 0.05
q1.2 <- qchisq(p = 1-alpha1, df = n, lower.tail = F)

# 5% de la probabilidad
alpha2 <- .1 - alpha1
q2.2 <- qchisq(p = 1-alpha2, df = n)

c(q1.2, q2.2)
```

```
## [1]  3.940299 18.307038
```

Por tanto, dado que el intervalo de confianza calculado en i) es $\big(\overline{X}_{10} - \sqrt{\frac{1}{250}q_2},\ \overline{X}_{10} - \sqrt{\frac{1}{250}q_1}\big)$, el intervalo con los valores elegidos de $q_{1.2}$ y $q_{2.2}$ sería:

```
CI.2 <- c(sum((muestra-1)^2)/q2.2, sum((muestra-1)^2)/q1.2)
CI.2
```

```
## [1] 0.01560506 0.07250273
```

Es decir:

$$CI.2 = \big(0.0156051,\ 0.0725027\big)$$