Práctica 3: Bootstrap semiparamétrico

Grupo 2: Martos Dourado Oscar Portela Vázquez Guillermo

TR 2024/2025

Esta práctica debe entregarse en formato pdf, incluyendo el código fuente utilizado, las correspondientes salidas y los comentarios (o interpretaciones de los resultados) pertinentes (para ello se recomienda emplear RMarkdown, a partir de un fichero *.Rmd* o un fichero *.R* mediante spin, que también debe entregarse).

Se debe establecer la semilla igual al número de grupo multiplicado por 10 (también se recomienda hacerlo antes de cada nueva generación de números pseudoaleatorios).

En esta práctica se empleará el conjunto de datos `Prestige` de la librería `carData`, considerando como variable respuesta `prestige` (puntuación de ocupaciones obtenidas a partir de una encuesta) y como variables explicativas: `income` (media de ingresos en la ocupación) y `education` (media de los años de educación).

Como punto de partida consideramos un modelo lineal:

```
library(carData)
```

```
## Warning: package 'carData' was built under R version 4.3.2
```

```
modelo <- lm(prestige ~ income + education, data = Prestige)
res <- summary(modelo)
res
```

```
##
## Call:
## lm(formula = prestige ~ income + education, data = Prestige)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.4040  -5.3308   0.0154   4.9803  17.6889
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.8477787  3.2189771  -2.127   0.0359 *
## income       0.0013612  0.0002242   6.071 2.36e-08 ***
## education    4.1374444  0.3489120  11.858  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 99 degrees of freedom
## Multiple R-squared:  0.798,  Adjusted R-squared:  0.7939
## F-statistic: 195.6 on 2 and 99 DF,  p-value: < 2.2e-16
```

Ejercicio 1

En primer lugar, supongamos que estamos interesados en realizar inferencias sobre la varianza del error. Podemos estimarla mediante la varianza residual:

$$\hat{S}^2_R = \frac{1}{n-p-1}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

```
rvar <- res$sigma^2      # with(modelo, sum(residuals^2)/df.residual)
rvar
```

```
## [1] 60.99849
```

Bajo las hipótesis estructurales del modelo, también podemos obtener estimaciones por intervalo de confianza:

$$IC_{(1-\alpha)}\left(\sigma^2\right) = \left(\frac{(n-p-1)\hat{S}^2_R}{\chi^2_{n-p-1,1-\alpha/2}}, \frac{(n-p-1)\hat{S}^2_R}{\chi^2_{n-p-1,\alpha/2}}\right)$$

```
alpha <- 0.05
rdf <- res$df[2]
cint <- rdf*rvar/qchisq(c(1 - alpha/2, alpha/2), df = rdf)
cint
```

```
## [1] 47.02350 82.31682
```

Alternativamente podríamos emplear bootstrap.

Utilizar la función `Boot()` del paquete `car` para obtener una estimación por intervalo de confianza de la varianza del error del modelo de regresión lineal (`prestige ~ income + education`) mediante *remuestreo residual*, empleando el método *percentil directo*.

```
library(tictoc)
```

```
## Warning: package 'tictoc' was built under R version 4.3.2
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.2
```

```
f_var <- function(obj) {
  with(obj,
       sum(residuals^2)/df.residual)
}

f_var(modelo) # función que otorga la varianza de un modelo
```

```
## [1] 60.99849
```

```
set.seed(20)
tic()
mod.boot <- Boot(modelo, method = "residual", f = f_var, labels = "varianza")
```

```
## Loading required namespace: boot
```

```
toc()
```

```
## 0.85 sec elapsed
```

```r
summary(mod.boot)
```

```
##            R original bootBias bootSE bootMed
## varianza 999   60.998  0.50145 8.1785  61.095
```

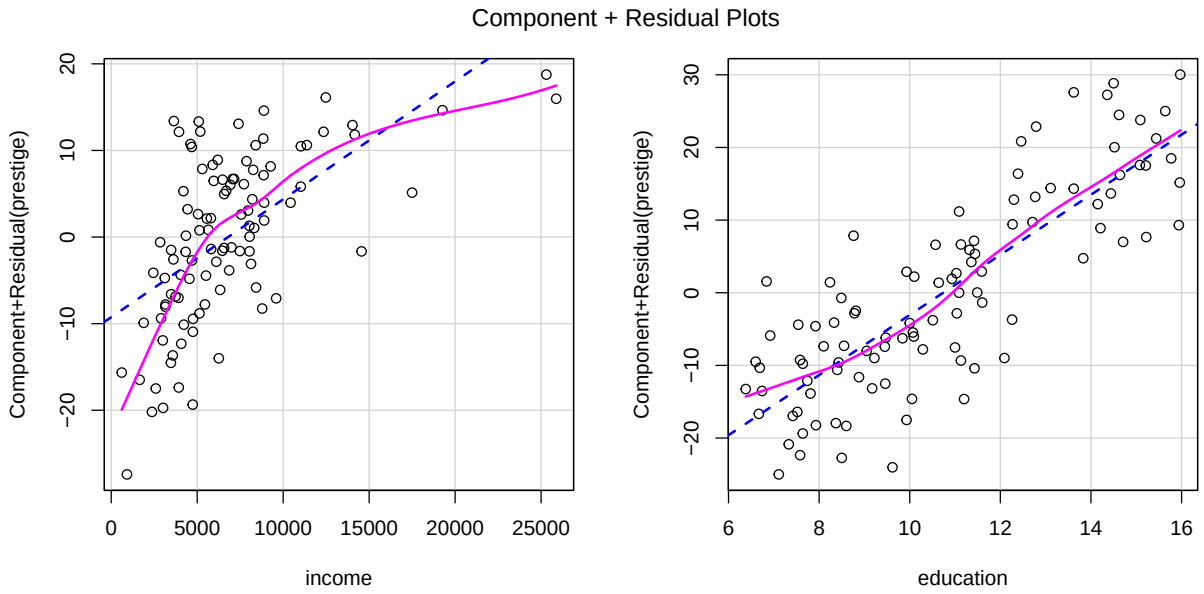
```r
round(
  confint(mod.boot, level = 0.95, type = "perc"),
  5)
```

```
## Bootstrap percent confidence intervals
##
##              2.5 %   97.5 %
## varianza 46.34396 79.58879
```

Obtenemos un intervalo más estrecho. ### Ejercicio 2

En segundo lugar, al estudiar el efecto de las variables explicativas en el modelo anterior podríamos pensar que no es adecuado asumir un efecto lineal de alguna de ellas. Si generamos los gráficos parciales de residuos obtendríamos:

```r
library(car)
crPlots(modelo)
```



Por ejemplo, podríamos considerar un efecto no lineal de la variable `income` ajustando un modelo aditivo con el paquete `mgcv`:

```r
library(mgcv)
```

```
## Warning: package 'mgcv' was built under R version 4.3.3
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```r
modelo2 <- gam(prestige ~ s(income) + education, data = Prestige)
summary(modelo2)
```

```
##
## Family: gaussian
```

```
## Link function: identity
##
## Formula:
## prestige ~ s(income) + education
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.2240     3.7323   1.132    0.261
## education     3.9681     0.3412  11.630   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df    F p-value
## s(income) 3.58  4.441 13.6  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.825   Deviance explained = 83.3%
## GCV = 54.798  Scale est. = 51.8      n = 102
```

Para comparar el ajuste de este modelo respecto al anterior, podemos realizar un contraste empleando la función `anova()`:

```
anova(modelo, modelo2)
```

```
## Analysis of Variance Table
##
## Model 1: prestige ~ income + education
## Model 2: prestige ~ s(income) + education
##   Res.Df    RSS     Df Sum of Sq      F    Pr(>F)
## 1  99.00 6038.9
## 2  96.42 4994.6 2.5802    1044.3 7.8131 0.0002245 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Alternativamente podríamos emplear bootstrap, aunque si se quieren reescalar los residuos de un modelo `gam`, como no implementan un método `hatvalues()`, habrá que emplear `influence.gam()` (o directamente `modelo.gam$hat`).

Contrastar si el efecto de `income` es lineal mediante bootstrap residual, empleando como estadístico el incremento en la variabilidad residual con el modelo reducido (modelo lineal), $\tilde{F} = (RSS_0 - RSS)/RSS$, y remuestreando los residuos (reescalados preferiblemente) del modelo completo (modelo aditivo). Aproximar el nivel crítico del contraste y el valor que tendría que superar el estadístico para rechazar H_0 con un nivel de significación $\alpha = 0.1$.

Primero reescalamos los residuos. Como no podemos usar `hatvalues()` hacemos una función que los calcule:

```
#función para reescalar los residuos
res_reescalados <- function(residuos, hat) {
  sres <- residuos/sqrt(1 - hat)
  sres <- sres - mean(sres)
  return(sres)
}
```

A continuación la usamos para calcular los residuos del modelo

```
pres.dat <- Prestige
pres.dat$sres <- res_reescalados(residuos = residuals(modelo2), hat = modelo2$hat)
```

Definimos una función que dados dos modelos calcula

$$\tilde{F} = \frac{RSS_0 - RSS}{RSS}$$

```
f_resdif <- function(modeloGrande, modeloPequeño) {
  RSS_0 <- sum(residuals(modeloPequeño)^2)
  RSS <- sum(residuals(modeloGrande)^2)
  (RSS_0 - RSS)/RSS
}
```

La usamos dentro de la siguiente función que es la que vamos a remuestrear, generando así dos modelos en cada remuestra:

```
f_statistic <- function(data, i) {
  data$prestige <- mean(data$prestige) + data$sres[i]
  mod_lm <- lm(prestige ~ income + education, data = data)
  mod_gam <- gam(prestige ~ s(income) + education, data = data)

  f_resdif(mod_gam, mod_lm)
}
```

Y a continuación realizamos bootstrap:

```
library(boot)
```

```
##
## Attaching package: 'boot'
```

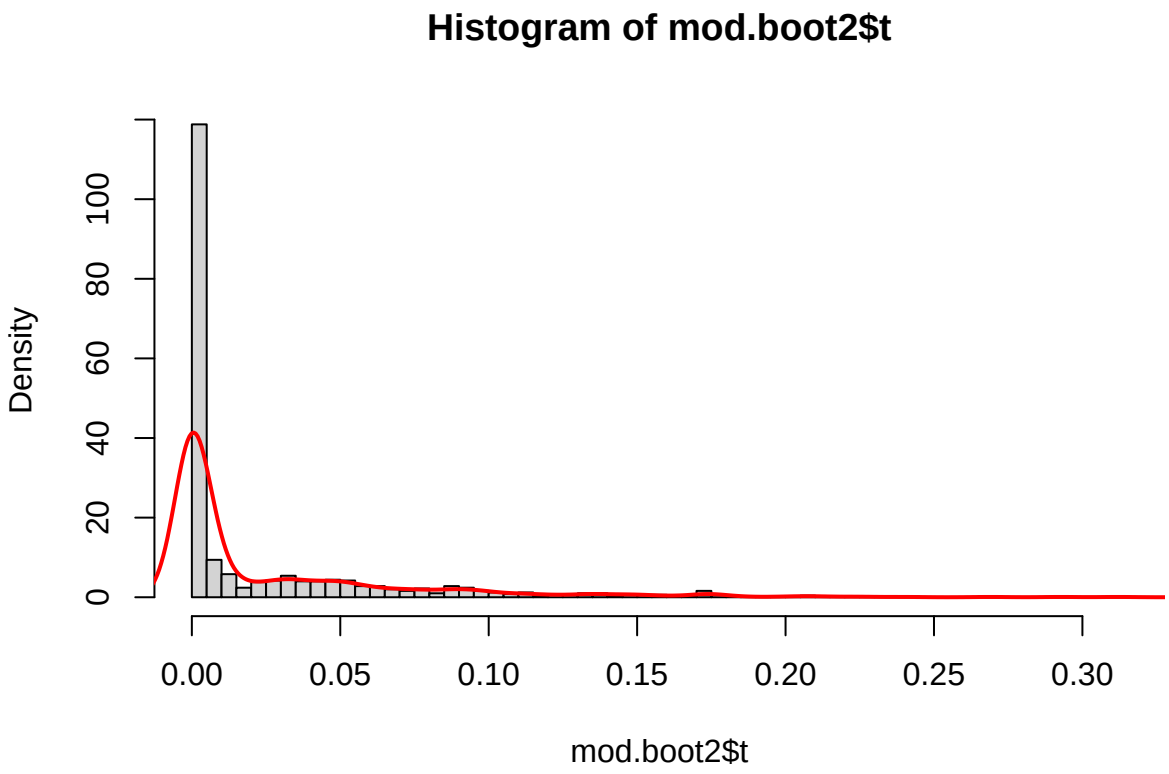
```
## The following object is masked from 'package:car':
##
##     logit
```

```
set.seed(20)
mod.boot2 <- boot(data = pres.dat, statistic = f_statistic, R = 1000)
summary(mod.boot2)
```

```
##      R   original bootBias   bootSE    bootMed
## 1 1000 1.0598e-12 0.025036 0.045028 1.5599e-11
```

Representando los valores obtenidos:

```
hist(mod.boot2$t, breaks = "FD", freq = FALSE)
lines(density(mod.boot2$t),
      col = "red",
      lwd = 2)
```



Donde se observa un problema de efecto frontera al usar density. Para aproximar el nivel crítico del contraste usamos el siguiente código:

```
pval <- mean(mod.boot2$t >= summary(modelo)$fstatistic[1])
pval
```

```
## [1] 0
```

La aproximación del valor que el estadístico tendría que superar para que se rechace la hipotesis nula (efecto lineal de income) con un nivel de significación de 0.1 corresponde con el valor que deja a la izquierda un 10% de los 1000 estadisticos $\tilde{F}^*$

```
quantile(mod.boot2$t, 0.9)
```

```
##        90%
## 0.08640901
```