Práctica 4 - Estadistica Computacional

Jaime Gomez Marin

## 2.- Calcular el error aparente, el error por validación cruzada 10 y el sesgo con sus interpretaciones para los datos Women (Y=2)

# numero de registros del dataset women  
dim(women)[1]

## [1] 15

### Cálculo del error aparente

APE=function(datos,y){  
 datos=as.matrix(datos)  
 n=dim(datos)[1]  
 resi=lm(datos[,y]~datos[,-y])$res  
 APE=sum(resi^2)/n  
 return(APE)  
}  
  
pos\_var\_dependiente = 2 # Y   
P = APE(women,pos\_var\_dependiente)  
P

## [1] 2.015556

### Cálculo del error por validación cruzada

Segun condición del problema, nos pide realizar una validacion cruzada con k=10, pero en el dataset “women” tenemos 15 registros, por lo tanto nuestro folk solo sera de floor(15/10) , es decir 1, por lo tanto nuestro fragmentos sera de 1, por lo tanto estamos hablando del caso particula de PRESS

# Calculo del tamaño de folk  
nro\_reg = dim(women)[1]  
K = 10  
floor(nro\_reg/K)

## [1] 1

#### PRESS

PRESS=function(datos,y) {  
 datos=as.matrix(datos)#convierte a matriz  
 n=dim(datos)[1]#num de filas  
 resi=rep(0,n)  
 for (i in 1:n) {  
 estim=sum(lm(datos[-i,y]~datos[-i,-y])$coe\*c(1,datos[i,-y]))#quita un dato y guardas coeficiente (betas)  
 resi[i]=datos[i,y]-estim  
 }  
 PRESS=sum(resi^2)/n  
 return(PRESS)  
}  
  
PRESS(women,2)

## [1] 3.040776

#### VALIDACION CRUZADA 10

Si deseamos usar validación cruzada, vamos a considerar que el tamaño del folk sea de 2 registros, por lo tanto K = 7 ( la mitad de la cantidad de registros del dataset women)

# Calculo del tamaño de folk  
nro\_reg = dim(women)[1]  
K = 7  
floor(nro\_reg/K)

## [1] 2

crossval = function(data,repet,K,y){  
 # repeticiones=repe  
 # k=particiones  
 data = as.matrix(data) # convertir en matriz a la data  
 n = dim(data)[1] #num filas de la matriz  
 p = dim(data)[2] #num de columnas de la matriz  
   
 EVC = rep(0, repet) # vector de ceros segun repeticiones  
   
 for (i in 1:repet) { # defnimos en el num repeticiones  
   
 resid = matrix(0,1,K) # vector de residuales de filas 1 y k columnas  
 indices = sample(1:n,n,replace=F) # n indices con reemplazo  
 azar = data[indices,] # seleccionar muestra (busca aleatorizar)  
 subm = floor(n/K) # redondea hacia abajo piso 48/10=4  
 #print(subm)  
 for (j in 1:K) {  
 #print(paste0("[i=",i,"][j=",j,"]"))  
 unid=((j-1)\*subm + 1):(j\*subm)  
 #print(paste0("[unid=",unid,"]"))  
 #print(unid)  
 if (j == K) {  
 unid=((j-1)\*subm+1):n  
 #print(paste0("[unid=====",unid,"]"))  
 #print(unid)  
 }  
 datap = azar[unid,]  
 datae = azar[-unid,]  
 ye = datae[,y]  
 xe = datae[,-y]  
 betas = lm(ye~xe)$coef  
 #print(datap)  
 #print(paste0("\*\*\*\*\*\*\*\*\*\*unos = ", dim(datap)))  
 unos = rep(1,dim(datap)[1])  
 #print(unos)  
 data1 = cbind(unos,datap[,-y])  
 predict = data1%\*%as.matrix(betas) # multiplicas matrices  
 resid[j] = sum((predict-datap[,y])^2) # residuales promedio  
 }  
   
 EVC[i]=sum(resid)/n  
 }  
   
 EVCP=mean(EVC)   
   
 return (list(EVC=EVC, EVCP=EVCP))  
}  
  
rep = 20  
K = 7  
pos\_var\_dependiente = 2 # Y   
crossval(women, rep, K, pos\_var\_dependiente)$EVCP

## [1] 3.015919

### Cálculo del sesgo

# Funcion para calcular el sesgo  
calc\_sesgo = function(datos, repet, K, Y) {  
 P = APE(datos,Y)  
 EVCP = crossval(datos, repet, K, Y)$EVCP  
 return(EVCP-P)  
}  
# Calculo del sesgo  
pos\_var\_dep = 2 # Y   
rep = 20  
Ks = c(2,3,4,5,6,7) # arreglo con todos los valor de K posibles  
  
for( K in Ks) {  
 sesgo = calc\_sesgo(women, rep, K, pos\_var\_dep)  
 print(paste0("K = ",K," => sesgo = ",sesgo))  
}

## [1] "K = 2 => sesgo = 1.53493155594429"  
## [1] "K = 3 => sesgo = 1.13766824228737"  
## [1] "K = 4 => sesgo = 1.40517582082843"  
## [1] "K = 5 => sesgo = 1.20570617675989"  
## [1] "K = 6 => sesgo = 0.899478206502136"  
## [1] "K = 7 => sesgo = 0.984810465588862"

Se puede observar que mayormente el sesgo se reduce conforme aumenta el valor de K

## 3.- Calcular los intervalos de confianza para los coeficientes de regresión por el método percentiles y por el método Estudentizado para los datos Data1, al 95% de confianza. Use B =150.

### Lectura de datos del archivo excel

#install.packages("xlsx")  
library(rJava)  
library(xlsx)  
library(xlsxjars)  
library(readxl)  
  
data.xls <- read.xlsx("Data1.xlsx", sheetIndex = 1,header=TRUE )  
str(data.xls)

## 'data.frame': 30 obs. of 7 variables:  
## $ Y : num 138 58 30 30 69 49 30 136 119 93 ...  
## $ X1: num 8 12 10 13 5 6 7 12 11 5 ...  
## $ X2: num 534 182 546 367 478 330 165 184 355 469 ...  
## $ X3: num 107.39 43.8 0.43 12.06 64.41 ...  
## $ X4: num 138.1 72.4 35.5 30.2 63 ...  
## $ X5: num 690 290 150 150 345 245 150 680 595 465 ...  
## $ X6: num 138.1 72.4 35.5 30.2 -63 ...

modelo = lm(Y~X1+X2+X3+X4+X5+X6, data=data.xls)  
summary(modelo)

## Warning in summary.lm(modelo): essentially perfect fit: summary may be  
## unreliable

##   
## Call:  
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6, data = data.xls)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.481e-13 -1.351e-14 4.252e-15 2.225e-14 4.805e-14   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.008e-14 3.744e-14 1.338e+00 0.1941   
## X1 5.355e-16 2.618e-15 2.050e-01 0.8397   
## X2 -1.294e-16 5.689e-17 -2.274e+00 0.0326 \*   
## X3 3.942e-16 3.429e-16 1.149e+00 0.2622   
## X4 0.000e+00 9.060e-16 0.000e+00 1.0000   
## X5 2.000e-01 1.963e-16 1.019e+15 <2e-16 \*\*\*  
## X6 -2.273e-16 8.949e-17 -2.540e+00 0.0183 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.939e-14 on 23 degrees of freedom  
## Multiple R-squared: 1, Adjusted R-squared: 1   
## F-statistic: 6.014e+30 on 6 and 23 DF, p-value: < 2.2e-16

Se observa que el coef. de regresión para las variables independientes X2, X5 y X6 son significativas.

### Definimos la función de bootstrap para calcular los coef.de regresión por el método de percentiles

ic.mp.boot.obser = function(datos,B,Y,nivel){  
 datos = as.matrix(datos)  
 alfa = 1-0.01\*nivel  
 n = dim(datos)[1]  
 c = ncol(datos)  
 betas = matrix(0,B,c)  
 for (i in 1:B){  
 indices = sample(1:n,n,T)  
 betas[i,] = lm(datos[indices,Y]~datos[indices,-Y])$coe  
 }  
 LI = apply(betas,2,quantile,alfa/2)  
 LS = apply(betas,2,quantile,1-alfa/2)  
 limites = cbind(LI,LS)  
 return(list(limites = limites))  
}  
  
index\_salida <- 1 # Posicion de la variable dependiente "Y"  
nro\_muestras\_bootstrap <- 150 # Nro. de muestras bootstrap  
nivel <- 95 # Nivel de confianza  
mp\_boot\_datos <- ic.mp.boot.obser(data.xls,nro\_muestras\_bootstrap,index\_salida, nivel)  
mp\_boot\_datos$limites

## LI LS  
## [1,] -1.115773e-13 5.814530e-14  
## [2,] -4.397748e-15 5.216316e-15  
## [3,] -7.718076e-17 1.264956e-16  
## [4,] -7.064176e-16 9.487051e-16  
## [5,] -1.524559e-15 2.193310e-15  
## [6,] 2.000000e-01 2.000000e-01  
## [7,] -1.675748e-16 1.733692e-16

#### Empleando el método de percentiles con Bootstrap con un nivel de significancia de 0.95, se tiene que:

#### El intervalo de confianza para el es entre [-1.115773510^{-13} , 5.814530210^{-14}]

#### El intervalo de confianza para el es entre [-4.397748110^{-15} , 5.216316510^{-15}]

#### El intervalo de confianza para el es entre [-7.718075610^{-17} , 1.264955710^{-16}]

#### El intervalo de confianza para el es entre [-7.064176210^{-16} , 9.487051310^{-16}]

#### El intervalo de confianza para el es entre [-1.524559210^{-15} , 2.193310510^{-15}]

#### El intervalo de confianza para el es entre [0.2 , 0.2]

#### El intervalo de confianza para el es entre [-1.67574810^{-16} , 1.73369210^{-16}]

### Definimos la función de bootstrap para calcular los coef.de regresión por el método de studentizados

ic.me.boot.obser = function(datos,B,Y,nivel) {  
 datos = as.matrix(datos)  
 alfa = 1-0.01\*nivel  
 n = dim(datos)[1]  
 c = ncol(datos)  
 coefi = lm(datos[,Y]~datos[,-Y])$coe  
 betas = matrix(0,B,c)  
 eebetas = matrix(0,B,c)  
 pivot = matrix(0,B,c)  
 for (i in 1:B){  
 indices = sample(1:n,n,T)  
 betas[i,] = lm(datos[indices,Y]~datos[indices,-Y])$coe  
 eebetas[i,] = summary(lm(datos[indices,Y]~datos[indices,-Y]))$coe[,2]  
 pivot[i,] = (betas[i,]-coefi)/eebetas[i,]  
 }  
 eebotbet = apply(betas,2,sd)  
 t1 = apply(pivot,2,quantile,alfa/2)  
 t2 = apply(pivot,2,quantile,1-alfa/2)  
 LI = coefi+t1\*eebotbet  
 LS = coefi+t2\*eebotbet  
 limites = cbind(LI,LS)  
   
 return(list(limites=limites))  
}  
  
index\_salida <- 1 # Posicion de la variable dependiente "Y"  
nro\_muestras\_bootstrap <- 150 # Nro. de muestras bootstrap  
nivel <- 95 # Nivel de confianza  
bootstrap\_datos <- ic.me.boot.obser(data.xls,nro\_muestras\_bootstrap,index\_salida, nivel)  
bootstrap\_datos$limites

## LI LS  
## (Intercept) -3.632739e-13 1.209349e-13  
## datos[, -Y]X1 -9.642331e-15 8.373963e-15  
## datos[, -Y]X2 -1.210771e-16 5.702316e-16  
## datos[, -Y]X3 -3.025691e-15 1.340491e-15  
## datos[, -Y]X4 -2.257900e-15 1.943292e-15  
## datos[, -Y]X5 2.000000e-01 2.000000e-01  
## datos[, -Y]X6 -1.508533e-16 7.250668e-16

#### Empleando el método de Estudentizados con Bootstrap con un nivel de significancia de 0.95, se tiene que:

#### El intervalo de confianza para el es entre [-3.632738510^{-13} , 1.209348710^{-13}]

#### El intervalo de confianza para el es entre [-9.642330810^{-15} , 8.373962910^{-15}]

#### El intervalo de confianza para el es entre [-1.210771410^{-16} , 5.702315610^{-16}]

#### El intervalo de confianza para el es entre [-3.025690810^{-15} , 1.340490910^{-15}]

#### El intervalo de confianza para el es entre [-2.257899710^{-15} , 1.943292110^{-15}]

#### El intervalo de confianza para el es entre [0.2 , 0.2]

#### El intervalo de confianza para el es entre [-1.508532910^{-16} , 7.25066810^{-16}]

## 4.- Hacer una función en R que estime los coeficientes de regresión usando el método VC8. Pruebe su función con los datos Data1.

crossval\_coef = function(data,repet,K,y){  
 # repeticiones=repe  
 # k=particiones  
 data = as.matrix(data) # convertir en matriz a la data  
 n = dim(data)[1] # num filas de la matriz  
 p = dim(data)[2] # num de columnas de la matriz  
 n\_betas = p # num de betas  
 EVC = matrix(0,rep,n\_betas) # vector de ceros segun repeticiones  
 # print(EVC)  
 for (i in 1:repet) { # defnimos en el num repeticiones  
   
 coef\_folk = matrix(0,K,n\_betas) # vector de residuales de filas 1 y k columnas  
 indices = sample(1:n,n,replace=F) # n indices con reemplazo  
 azar = data[indices,] # seleccionar muestra (busca aleatorizar)  
 subm = floor(n/K) # redondea hacia abajo piso 48/10=4  
 # print(subm)  
 # print(coef\_folk)  
 for (j in 1:K) {  
 unid=((j-1)\*subm + 1):(j\*subm)  
 if (j == K)   
 unid=((j-1)\*subm+1):n  
 datap = azar[unid,]  
 datae = azar[-unid,]  
 ye = datae[,y]  
 xe = datae[,-y]  
 betas = lm(ye~xe)$coef  
 matriz\_betas = as.matrix(betas)   
 coef\_folk[j,] = matriz\_betas  
 # print(coef\_folk)  
 }  
 EVC[i,]=apply(coef\_folk,2,mean)  
 # print(coef\_folk)  
 # print(EVC)  
  
 }  
 # print(coef\_bootstrap)  
   
 EVCP=apply(EVC,2,mean)   
   
 return (list(EVC=EVC, EVCP=EVCP))  
}  
  
  
rep = 20 # num de repeticiones bootstrap  
K = 8 # VC8   
pos\_var\_dependiente = 1 # Posicion de la variable dependiente "Y"  
BETAS = crossval\_coef(data.xls, rep, K, pos\_var\_dependiente)$EVCP  
BETAS

## [1] -5.115954e-16 5.121925e-18 -2.467510e-18 -7.626180e-17 7.124499e-17  
## [6] 2.000000e-01 2.771648e-18

#### es [-5.115954410^{-16}]

#### es [5.121924610^{-18}]

#### es [-2.467509810^{-18}]

#### es [-7.626180410^{-17}]

#### es [7.124498510^{-17}]

#### es [0.2]

#### es [2.77164810^{-18}]