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 (b
    resi[i]=datos[i,y]-estim
  }
  PRESS=sum(resi^2)/n
  return(PRESS)
}
```

```r
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)

```r
# Calculo del tamaño de folk
nro_reg = dim(women)[1]
K = 7
floor(nro_reg/K)
```

```
## [1] 2
```

```r
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] 3.017217
```

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 = 1.98596778995735"
## [1] "K = 3 => sesgo = 1.2444714292897"
## [1] "K = 4 => sesgo = 1.36698968376552"
## [1] "K = 5 => sesgo = 1.24920621246385"
## [1] "K = 6 => sesgo = 1.27210019817293"
## [1] "K = 7 => sesgo = 1.17460443970877"
```

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)
```

```r
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 ...
```

```r
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

```r
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,] -1.333701e-13 7.849960e-14
## [2,] -5.146565e-15 5.190248e-15
## [3,] -1.169212e-16 9.697846e-17
## [4,] -9.583355e-16 8.917113e-16
## [5,] -1.497848e-15 1.475548e-15
## [6,]  2.000000e-01 2.000000e-01
## [7,] -1.721769e-16 1.591195e-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 β_0 es entre $[-1.3337009 \times 10^{-13}$, $7.8499603 \times 10^{-14}]$

El intervalo de confianza para el β_1 es entre $[-5.1465649 \times 10^{-15}$, $5.1902482 \times 10^{-15}]$

El intervalo de confianza para el β_2 es entre $[-1.169212 \times 10^{-16}$, $9.6978455 \times 10^{-17}]$

El intervalo de confianza para el β_3 es entre $[-9.5833552 \times 10^{-16}$, $8.9171128 \times 10^{-16}]$

El intervalo de confianza para el β_4 es entre $[-1.4978477 \times 10^{-15}$, $1.4755477 \times 10^{-15}]$

El intervalo de confianza para el β_5 es entre $[\mathbf{0.2}$, $\mathbf{0.2}]$

El intervalo de confianza para el β_6 es entre $[-1.7217689 \times 10^{-16}$, $1.5911946 \times 10^{-16}]$

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)
```

```r
  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)   -3.649644e-13 9.180824e-14
## datos[, -Y]X1 -8.783850e-15 6.799941e-15
## datos[, -Y]X2 -9.228987e-17 5.483609e-16
## datos[, -Y]X3 -4.014286e-15 1.334902e-15
## datos[, -Y]X4 -1.933243e-15 1.638178e-15
## datos[, -Y]X5  2.000000e-01 2.000000e-01
## datos[, -Y]X6 -2.065737e-16 7.859921e-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 β_0 es entre $[-3.6496443 \times 10^{-13}$ **,** $9.1808239 \times 10^{-14}]$

El intervalo de confianza para el β_1 es entre $[-8.7838497 \times 10^{-15}$ **,** $6.7999413 \times 10^{-15}]$

El intervalo de confianza para el β_2 es entre $[-9.2289875 \times 10^{-17}$ **,** $5.4836088 \times 10^{-16}]$

El intervalo de confianza para el β_3 es entre $[-4.0142865 \times 10^{-15}$ **,** $1.3349023 \times 10^{-15}]$

El intervalo de confianza para el β_4 es entre $[-1.9332434 \times 10^{-15}$ **,** $1.6381781 \times 10^{-15}]$

El intervalo de confianza para el β_5 es entre [0.2 , 0.2]

El intervalo de confianza para el β_6 es entre $[-2.065737 \times 10^{-16}$ **,** $7.8599209 \times 10^{-16}]$

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

```r
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] -2.477120e-17 -2.550211e-16  1.084737e-17  1.946913e-18  7.301906e-19
## [6]  2.000000e-01 -2.059027e-18
```

β_0 **es** $[-2.4771203 \times 10^{-17}]$

β_1 **es** $[-2.5502112 \times 10^{-16}]$

β_2 **es** $[1.084737 \times 10^{-17}]$

β_3 **es** $[1.9469125 \times 10^{-18}]$

β_4 **es** $[7.3019061 \times 10^{-19}]$

β_5 **es** $[\mathbf{0.2}]$

β_6 **es** $[-2.0590272 \times 10^{-18}]$