Computational Statistics. Chapter 6: Bootstrapping. Solution of exercises

Thierry Denoeux

2023-01-31

```
set.seed(20211015)
```

Exercise 1

Question a

Data generation:

```
n<-50
rate<-0.5
x<-rexp(n,rate=rate)
```

The cdf of the exponential distribution is $F(x) = 1 - \exp(-\theta x)$. The equation $F(m) = 0.5$ gives us $m = \log(2)/\theta$ Here, we get

```
mtrue<-log(2)/rate
print(mtrue)
```

```
## [1] 1.386294
```

Question b

```
bootstrap <- function(x,B){
    n<-length(x)
    X<-matrix(0,B,n)
    for(b in (1:B)) X[b,]<-sample(x,size=n,replace=TRUE)
    return(X)
}
```

Question c

The MLE of θ is $\widehat{\theta} = 1/\overline{X}$, where $\overline{X}$ is the sample mean. The plug-in estimate of m is

$$\widehat{m} = \log(2)/\widehat{\theta} = \log(2)\overline{X}.$$

We write the following R function:

```r
median_plug<-function(x) return(mean(x)*log(2))
```

For our dataset, we get

```r
mhat<-median_plug(x)
print(mhat)
```

```
## [1] 1.552819
```

To estimate the standard error of $\widehat{m}$, we generate B bootstrap samples and we compute the standard deviation of the bootstrap estimates:

```r
B<- 1000
X<-bootstrap(x,B)  # generation of B bootstrap samples
med<-apply(X,1,median_plug)  # calculation of the B bootstrap estimates
se_B<- sd(med)  # calculation of the standard deviation
print(se_B)
```

```
## [1] 0.1806734
```

Question d

The bounds of the $1-\alpha$ bootstrap percentile confidence interval are simply the $\alpha/2$ anf $1-\alpha/2$ empirical quantiles of the B bootstrap estimates:

```r
alpha<- 0.05
CI1<-quantile(med,c(alpha/2,1-alpha/2))
print(CI1)
```

```
##     2.5%    97.5%
## 1.221340 1.912688
```

Question e

To compute bootstrap estimates of the standard error of $\widehat{m}$, we need to bootstrap each bootstrap sample. We will use only $B_1 = 50$ pseudo-datasets in the inner bootstrap loop:

```r
B1<-50
se<-vector("numeric",B)
for(b in 1:B){
    X1<- bootstrap(X[b,],B1)  # Bootstrapping bootstrap sample b
    med_star<-apply(X1,1,median_plug)
    se[b]<- sd(med_star)
    }
```

We can then compute the quantiles of the approximately pivotal quantity

$$Z^* = \frac{\widehat{m}^* - \widehat{m}}{\mathsf{se}^*}$$

```r
q<-quantile((med-mhat)/se,c(alpha/2,1-alpha/2))
CI2<- c(mhat-q[2]*se_B,mhat-q[1]*se_B)
print(CI2)
```

```
##    97.5%     2.5%
## 1.213185 2.022918
```

Question f

To estimate the coverage probabilities of the two confidence intervals above, we will generate $N = 200$ datasets and compute the corresponding confidence intervals. The coverage probability is estimated by the proportion of intervals containing the true value of the parameter.

```
N<-200
k1<-0
k2<-0
for(i in 1:N){
  x<-rexp(n,rate=rate)  # Sample generation
  X<-bootstrap(x,B)  # bootstrapping
  med<-apply(X,1,median_plug)  # computation of the plug-in estimates
  CI1<-quantile(med,c(alpha/2,1-alpha/2))  # percentile confidence interval
  if((CI1[1]<=mtrue)&(mtrue<=CI1[2])) k1<-k1+1  # k1 is incremented if the CI contains the true median
  mhat<-median_plug(x)
  for(b in 1:B){
    X1<- bootstrap(X[b,],B1)  # Bootstrapping bootstrap sample b
    med_star<-apply(X1,1,median_plug)
    se[b]<- sd(med_star)
  }
  q<-quantile((med-mhat)/se,c(alpha/2,1-alpha/2))
  CI2<- c(mhat-q[2]*se_B,mhat-q[1]*se_B)  # Bootstrap t interval
  if((CI2[1]<=mtrue)&(mtrue<=CI2[2])) k2<-k2+1
}
p1<-k1/N
p2<-k2/N
print(c(p1,p2))
```

```
## [1] 0.955 0.960
```

The two confidence intervals appear to have similar coverage probabilities.

Exercise 2

Question a

We first read and transform the data, and create a new data frame:

```
data <- read.table("/Users/Thierry/Documents/R/Data/Compstat/investment.txt", header=TRUE)
y<- data$Invest/(data$CPI*10)
time<- 1:15
GNP1<-data$GNP/(data$CPI*10)
data1<-data.frame(Invest=y,time, GNP=GNP1,data$Interest,data$Inflation)
```

We then perform linear regression on this dataset using function `lm`, and compute 95% confidence intervals on the coefficients under the assumptions of the standard linear regression model:

```
reg<- lm(Invest~.,data=data1)
CI.norm<-confint(reg, level = 0.95)
print(CI.norm)
```

```
##                      2.5 %        97.5 %
## (Intercept)    -0.629244578 -0.3888960778
## time           -0.020888726 -0.0122906337
```

```
## GNP             0.550432484  0.7901765887
## data.Interest  -0.005086471  0.0002303046
## data.Inflation -0.002874148  0.0030021973
```

Question b

We first load package `boot`:

```
library("boot")
```

We then write the function for the calculation of the bootstrapped statistics, to be provided to function `boot`. The first argument of this function must be the dataset, and the second argument must be a vector of indices of any bootstrap sample for which the statistics will be computed:

```
reg_invest<-function(data1,ii){
  reg<- lm(Invest~.,data=data1[ii,])
  return(reg$coefficients)
}
```

We then call function `boot` and compute the percentile and BC_a confidence intervals using function `boot.ci`:

```
bt<- boot(data1, reg_invest, R=1000)
CI.cases.perc<-matrix(0,5,2)
CI.cases.bca<-matrix(0,5,2)
for(j in 1:5){
  CI<-boot.ci(bt, conf = 0.95, index=j,type = c("perc","bca"))
  CI.cases.perc[j,]=CI$percent[4:5]
  CI.cases.bca[j,]=CI$bca[4:5]
}
```

We print the two sets of confidence intervals side by side:

```
print(cbind(CI.cases.perc,CI.cases.bca))
```

```
##              [,1]          [,2]         [,3]         [,4]
## [1,] -0.584873626 -3.464810e-01 -0.614060368 -0.387058999
## [2,] -0.019815656 -1.017082e-02 -0.020481954 -0.011126080
## [3,]  0.508326461  7.456245e-01  0.540327530  0.775249668
## [4,] -0.007239239  5.840694e-05 -0.006273666  0.001410346
## [5,] -0.004436389  5.762746e-03 -0.005331037  0.005017292
```

Question c

We first compute the residuals:

```
fit<- fitted(reg)
residuals<-residuals(reg)
```

We put the design matrix in a matrix `X` that will be useful later:

```
X<-model.matrix(reg)
```

The following function computes the statistics that will be passed to function `boot`:

```
reg_invest1<-function(residuals,ii,fit,X){
  y<-fit+residuals[ii]  # construction of pseudo-responses
  reg<- lm(y~X-1) # regression on the pseudo responses
```

```
  return(reg$coefficients)
}
```

Finally, we compute the bootstrap percentile and BC_a confidence intervals by passing function `reg_invest1` to `boot` and then using function `boot.ci`:

```
bt1<- boot(residuals, reg_invest1, R=1000,fit=fit,X=X)
CI.res.perc<-matrix(0,5,2)
CI.res.bca<-matrix(0,5,2)
for(j in 1:5){
  CI<-boot.ci(bt1, conf = 0.95, index=j,type = c("perc","bca"))
  CI.res.perc[j,]=CI$percent[4:5]
  CI.res.bca[j,]=CI$bca[4:5]
}
```

We print the two sets of confidence intervals side by side:

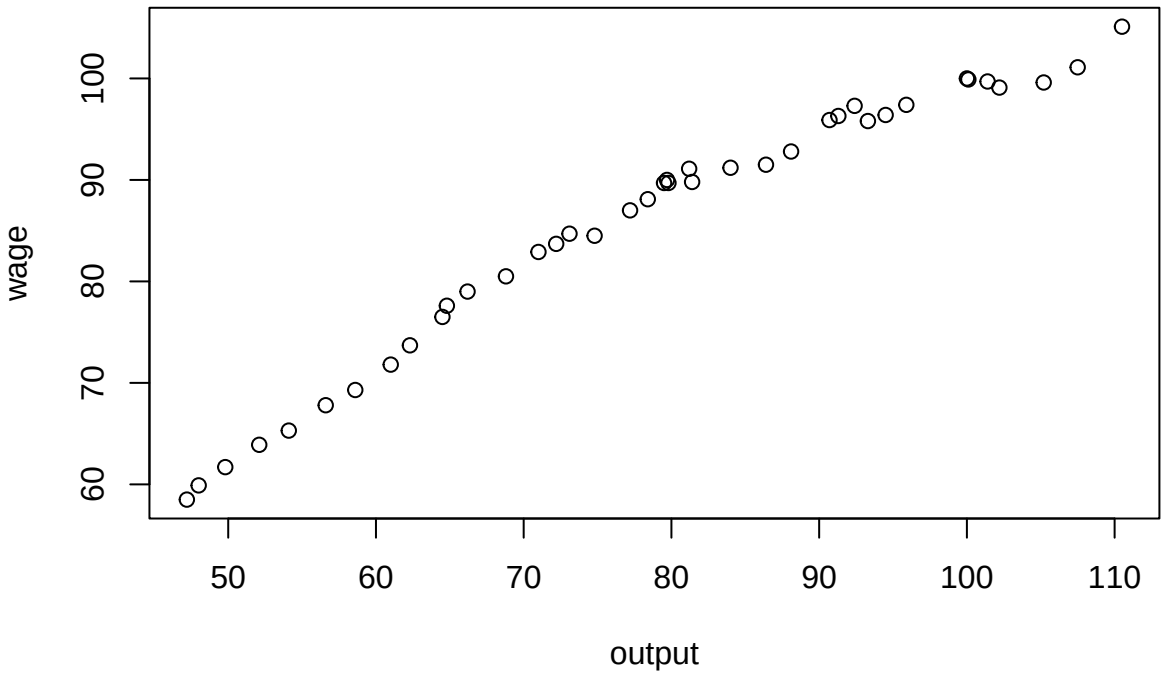
```
print(cbind(CI.res.perc,CI.res.bca))
```

```
##              [,1]          [,2]         [,3]          [,4]
## [1,] -0.600590910 -0.4232504123 -0.602117342 -0.4267097343
## [2,] -0.020005544 -0.0135823636 -0.020005447 -0.0135803342
## [3,]  0.587346489  0.7639053914  0.587782312  0.7658976891
## [4,] -0.004372466 -0.0005305074 -0.004518201 -0.0005871002
## [5,] -0.002059952  0.0021505491 -0.002306777  0.0019607758
```

Exercise 3

Question a

```
data <- read.table("/Users/Thierry/Documents/R/Data/Compstat/wages.txt",header=TRUE)
attach(data)
plot(output,wage)
```



Question b

Computing the LS estimates and the normal theory confidence intervals:

```
reg<- lm(wage~output+I(output^2)) # Least-squares estimation
summary(reg)
```

```
##
## Call:
## lm(formula = wage ~ output + I(output^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.58580 -0.76248  0.09209  0.68442  2.63570
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.622e+01  2.955e+00  -5.489 3.09e-06 ***
## output       1.949e+00  7.799e-02  24.987  < 2e-16 ***
## I(output^2) -7.917e-03  4.968e-04 -15.936  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9669 on 37 degrees of freedom
## Multiple R-squared:  0.9947, Adjusted R-squared:  0.9944
## F-statistic:  3483 on 2 and 37 DF,  p-value: < 2.2e-16
```

```
CI1<-confint(reg,  level = 0.95)
print(CI1)
```

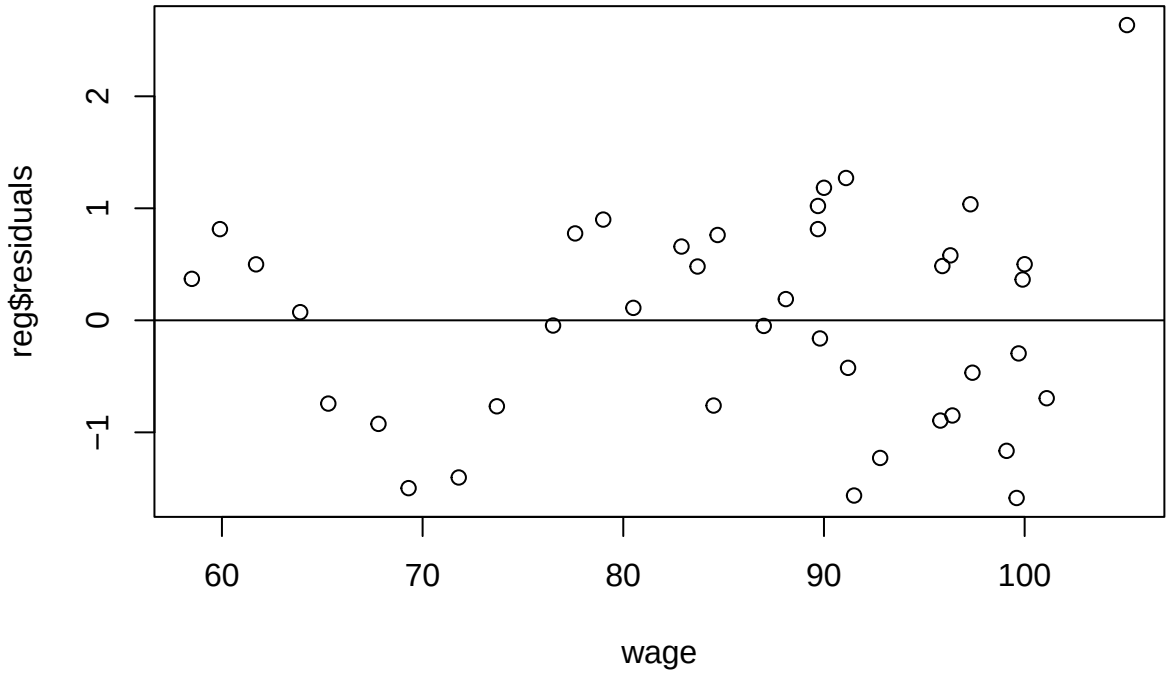
```
##                    2.5 %        97.5 %
## (Intercept) -22.2047470 -10.231601492
```

```
## output         1.7907987   2.106861776
## I(output^2)   -0.0089231  -0.006910034
```

Question c

Plotting the residuals:

```
plot(wage,reg$residuals)
abline(h=0)
```



Loading package `lmtest` and running the Durbin-Watson test:

```
library("lmtest")
```

```
## Le chargement a nécessité le package : zoo
```

```
##
## Attachement du package : 'zoo'
```

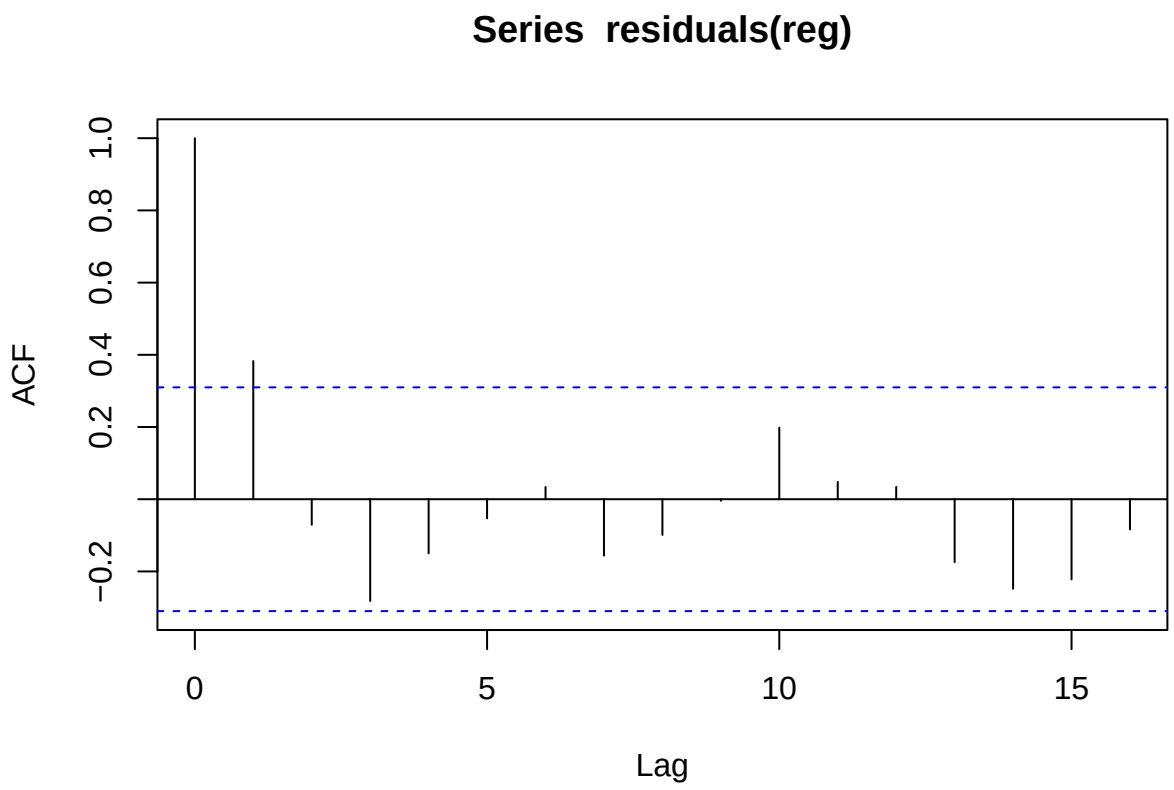
```
## Les objets suivants sont masqués depuis 'package:base':
##
##     as.Date, as.Date.numeric
```

```
dwtest(reg)
```

```
##
##  Durbin-Watson test
##
## data:  reg
## DW = 1.03, p-value = 0.0001178
## alternative hypothesis: true autocorrelation is greater than 0
```

We reject the null hypothesis of zero autocorrelation. This is confirmed by the autocorrelation plot:

```
acf(residuals(reg))
```



Question d

Preparing the data to be passed to function `tsboot`:

```
fit<- fitted(reg)  # the fitted values
residuals<-ts(residuals(reg)) # converting the residuals into an object of type "time-series"
X<-model.matrix(reg) # design matrix
```

The following function computes the statistics that will be passed to `tsboot`:

```
coef_wages<-function(residuals,fit,X){
  y<-fit+residuals
  reg<- lm(y~X-1)
  return(reg$coefficients)
}
```

Calling function `tsboot` and computing the bootstrap percentile confidence intervals on the coefficients:

```
bt2<- tsboot(residuals, coef_wages, R=999, l = 6, sim = "fixed",fit=fit,X=X)
CI2<-matrix(0,3,2)
CI2[1,]<-boot.ci(bt2, conf = 0.95, index=1,type = c("perc"))$percent[4:5]
CI2[2,]<-boot.ci(bt2, conf = 0.95, index=2,type = c("perc"))$percent[4:5]
CI2[3,]<-boot.ci(bt2, conf = 0.95, index=3,type = c("perc"))$percent[4:5]
```

Printing the two sets of confidence intervals side by side:

```
print(cbind(CI1,CI2))
```

```
##                   2.5 %        97.5 %
```

```
## (Intercept) -22.2047470 -10.231601492 -22.590910872 -10.141873829
## output        1.7907987   2.106861776   1.787960593   2.117594785
## I(output^2)  -0.0089231  -0.006910034  -0.009013346  -0.006883641
```

The bootstrap confidence intervals are slightly wider than the normal theory ones, which are based on wrong assumptions.