Bootstrap_correction

Muyao GUO

2025-10-10

```r
rm(list=objects())      # supprime les objets existant en session
graphics.off()          # supprime les graphiques existant en session
#setwd("...")            # définit le répertoire en cours

#######################################################

# Partie 1
#######################################################

x=c(3.1,2.4,2.6,2.2,1.9,2.8,1.1,0.7,2.3,4.3)
n=length(x)

# Loi de Rayleigh
library(VGAM) # utilisation de drayleigh
```

```
## 载入需要的程序包：stats4
```

```
## 载入需要的程序包：splines
```

```r
curve(drayleigh(x,scale=2),from=0,to=10)

# calcul de l'EMV via optimize
logL=function(a) sum(log(drayleigh(x,scale=a)))
opt=optimize(logL, interval = c(0,6), maximum = TRUE)
print(opt)
```

```
## $maximum
## [1] 1.787453
##
## $objective
## [1] -14.1634
```

```r
opt$maximum # [1] 1.787453
```

```
## [1] 1.787453
```

```r
# calcul de l'EMV via optim
nlogL=function(a) -sum(log(drayleigh(x,scale=a))) # minimisation!
res=optim(par=1,fn=nlogL,method="BFGS",hessian=T)
```

```
## Warning in log(scale[xok]): 产生了NaNs
```

```r
res$par  # idem qu'avec optim
```

```
## [1] 1.787436

res$hessian # permet le calcul de l'info de Fisher

##          [,1]
## [1,] 12.52032

# calcul de l'EMV par dérivation de logL
a_est=sqrt(mean(x^2)/2) # pourquoi? faire le calcul
a_est

## [1] 1.787456

# [1] 1.78932
Ia=4*n/a_est^2 # pourquoi? faire le calcul
# [1] [1] 12.51956 (à comparer à sa valeur approchée res$hessian =
12.52032)

# intervalle de confiance
c(a_est-qnorm(.975)/sqrt(Ia),a_est+qnorm(.975)/sqrt(Ia))

## [1] 1.233528 2.341385

# [1] 1.233528 2.341385
# construction à savoir expliquer!

############### bootstrap
set.seed(123)
B=1000

nlogL=function(a,ech) -sum(log(drayleigh(ech,scale=a)))
a_est=optim(par=1,fn=nlogL,ech=x,method="BFGS")$par  # idem que ci-
dessus

## Warning in log(scale[xok]): 产生了NaNs

a_boot=replicate(B,optim(par=1,fn=nlogL,ech=sample(x,n,replace=TRUE),me
thod="BFGS")$par)

length(a_boot) # 1000

## [1] 1000

# Loi d'échantillonage bootstrap centrée
T_boot=a_boot-a_est
hist(T_boot)
# ici l'histogramme présente une allure gaussienne symétrique
# c'est attendu car on sait que la loi d'un EMV est approchée par la
loi gaussienne
# ce n'est pas toujours le cas dans d'autres problèmes (et c'est même
# l'intérêt!)
```

```r
mean(T_boot)  # estimation bootstrap du biais -0.005453728

## [1] -0.005453728

sd(T_boot)   # estimation bootstrap de l'écart-type  0.2108019

## [1] 0.2108019

MSE= mean(T_boot)^2 + var(T_boot)
MSE   # 0.04446719

## [1] 0.04446719

## IC bootstrap # à savoir justifier
ICb=data.frame(binf=as.numeric(a_est-quantile(T_boot,0.975)),
               bsup=as.numeric(a_est-quantile(T_boot,0.025)))
ICb

##       binf     bsup
## 1 1.386613 2.197432

#       binf     bsup
# 1 1.386613 2.197432

# sur ce run, l'IC bootstrap est plus court que l'IC fondé sur la loi
# approchée gaussienne
# pour conclure que l'IC bootstrap est plus précis (ou pas)
# il faudrait faire des simulations sur un nombre suffisant de run.
```