TP1 Bootstrapt

Muyao GUO

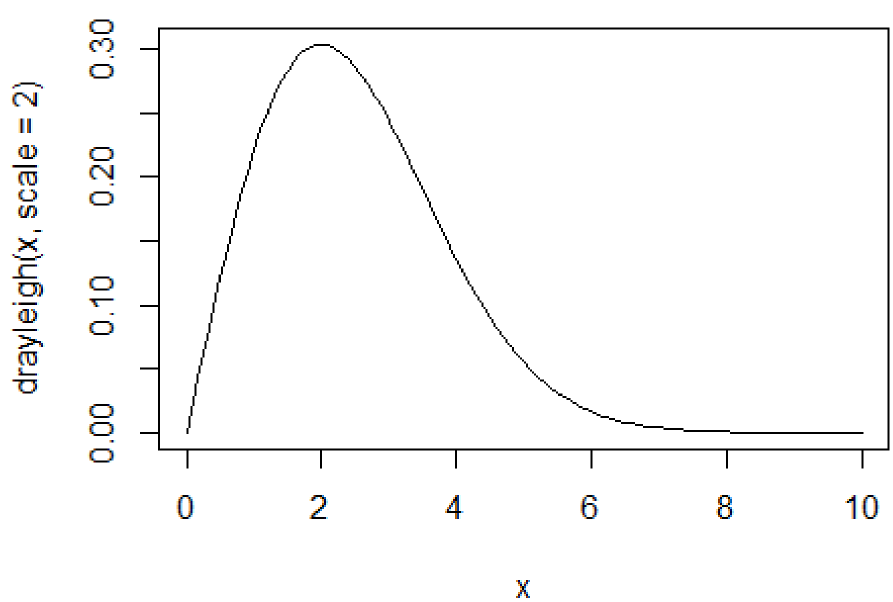
2025-09-11

TP1 Bootstrap

```
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

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



```r
# 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
```

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

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

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

```
## [1] 1.787456
```

```r
# [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)
```

通过手动计算二次求导的方法计算estimateur maximun vraissemblance

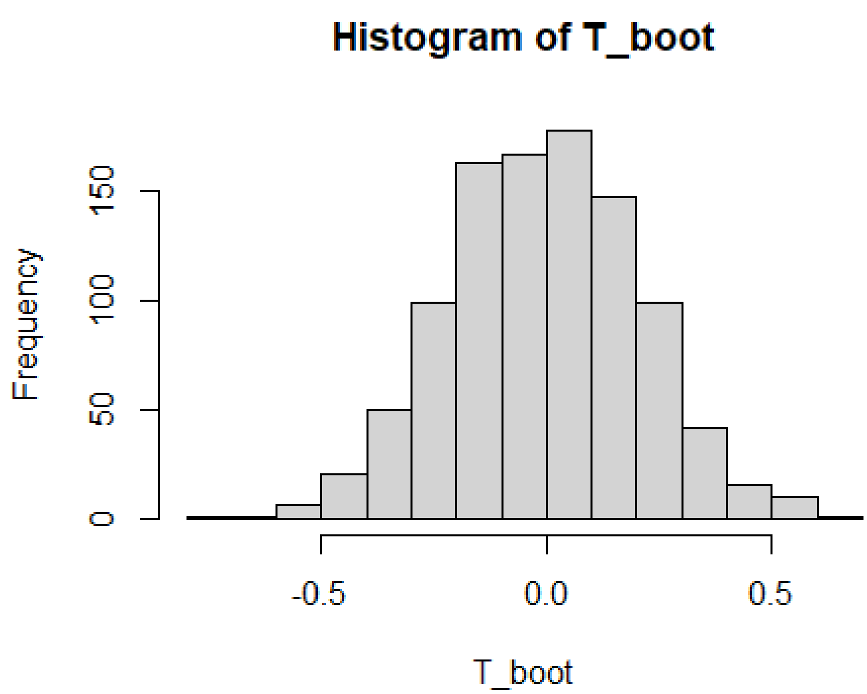
```r
# intervalle de confiance
c(a_est-qnorm(.975)/sqrt(Ia),a_est+qnorm(.975)/sqrt(Ia))
```

```
## [1] 1.233528 2.341385
```

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

**Bootstrapt模拟

```r
############### 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)
```

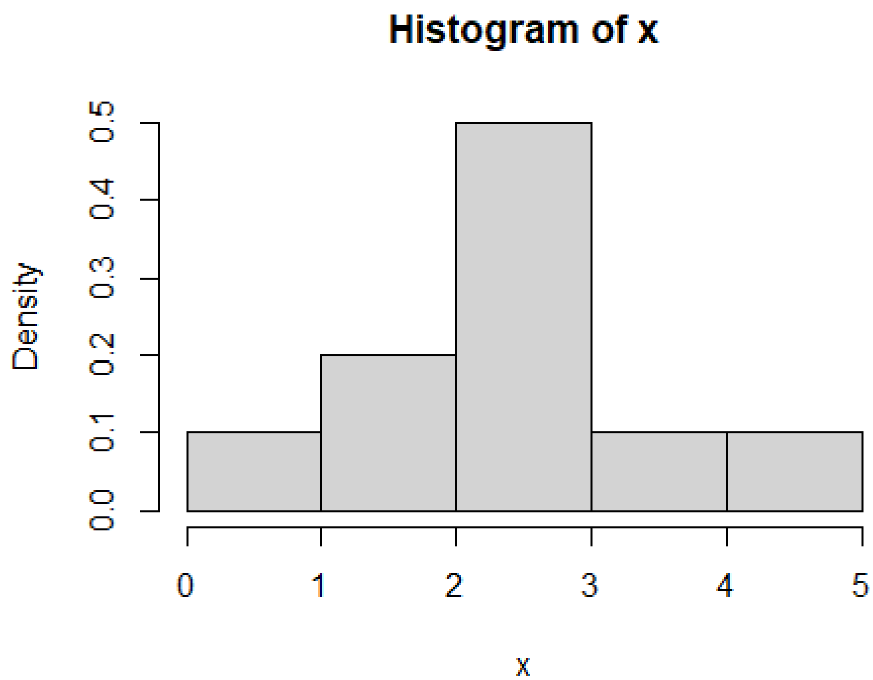



```r
# ici l'histogramme présente une allure gaussienne symétrique
# c'est attendu car on sait que la loi d'un EMV est approchée par la
loi gaussienne
# ce n'est pas toujours le cas dans d'autres problèmes (et c'est même
# l'intérêt!)
```

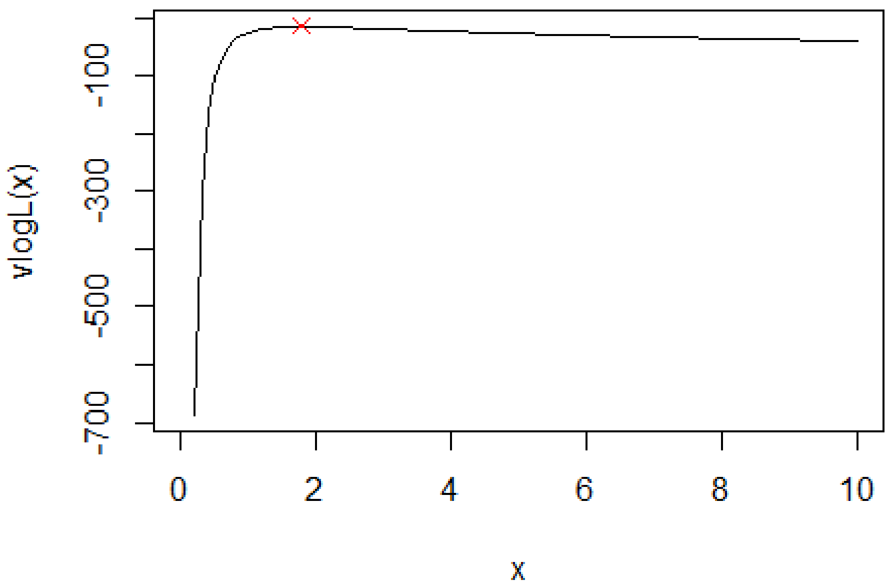
```r
mean(T_boot)  # estimation bootstrap du biais -0.005453728

## [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.
```

我的思路

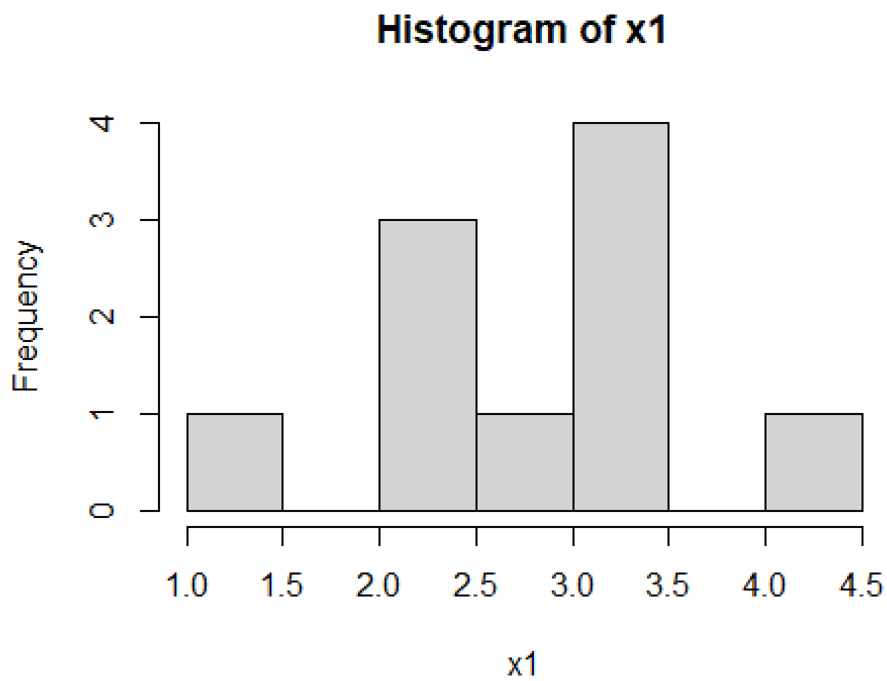
```r
library(VGAM)

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)
hist(x,prob=T)
```



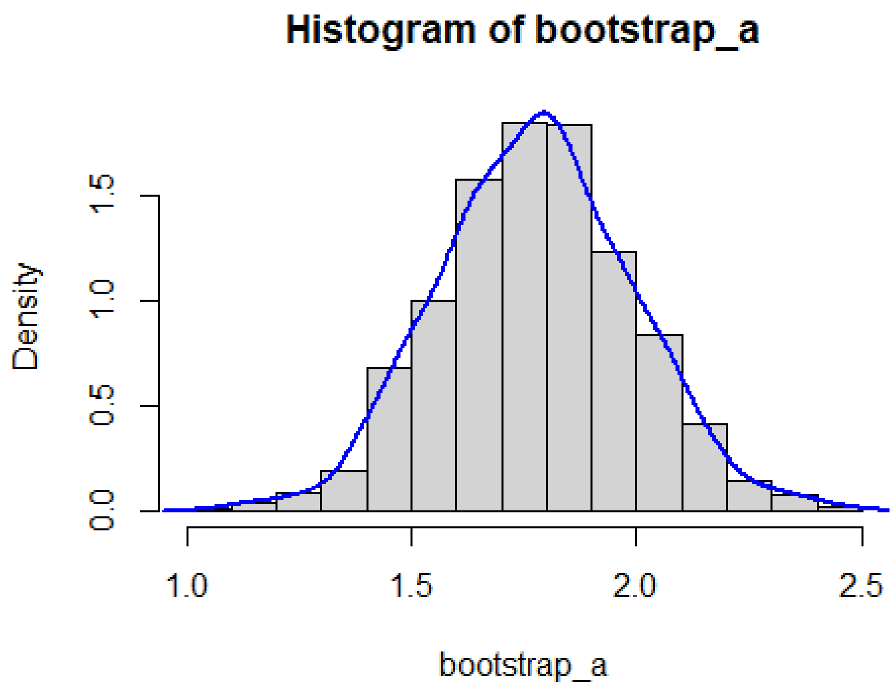

```r
# Histogram estimateur non paramerique

# Q1
# 用optimise 返回EMV
fdensite=function(x,a) x/(a**2)*exp(-x**2/2/(a**2))
logL = function(a) sum(log(fdensite(x=x,a=a)))
a_hat = optimize(logL,c(0.01,10),maximum = T)

vlogL = Vectorize(logL,"a")
curve(vlogL, from=0.01, to=10)
points(a_hat$maximum,a_hat$objective, col="red",pch = 4)
```




```
a_hat$maximum    #1.787453

## [1] 1.787453

# Q2
x1 = sample(x,10,replace = TRUE)
hist(x1)
```




```r
estimateur_1 = mean(x1)   # estimateur bootstrapt 1.82

# Q3
set.seed(1907)
B=1000
#help(replicate)

a_i_hat = function(){
  # (1) 重新抽样
  x_i = sample(x,n,replace = TRUE)
  # (2) 定义方程式
  fdensite=function(x,a) x/(a**2)*exp(-x**2/2/(a**2))
  logL = function(a) sum(log(fdensite(x=x_i,a=a)))
  # (3) 计算该样本的最大似然估计值
  a_hat = optimize(logL,c(0.01,10),maximum = T)
  return(a_hat$maximum)
}

# 1000个a_i_*_hat
bootstrap_a = replicate(B,a_i_hat())
hist(bootstrap_a,probability = T)
lines(density(bootstrap_a), col = "blue", lwd = 2)
```




```r
# Q4
a_origine = a_hat$maximum    #1.787453
MSE  = mean((bootstrap_a - a_origine)**2)
MSE   #0.04637799

## [1] 0.04637799

#或者用MSE = Biais^2(a_hat) - var(a_hat)
var_a_hat = var(bootstrap_a)
biais  = mean(bootstrap_a) - a_origine
MSE_2  = biais**2 + var_a_hat
MSE_2   # 0.04642429

## [1] 0.04642429

# Q5
#help("quantile")
IC = quantile(bootstrap_a, c(0.025,0.975))
IC   # 1.370749 2.193389

##      2.5%    97.5%
## 1.370749 2.193389
```