Solution lab 18

May 6, 2016

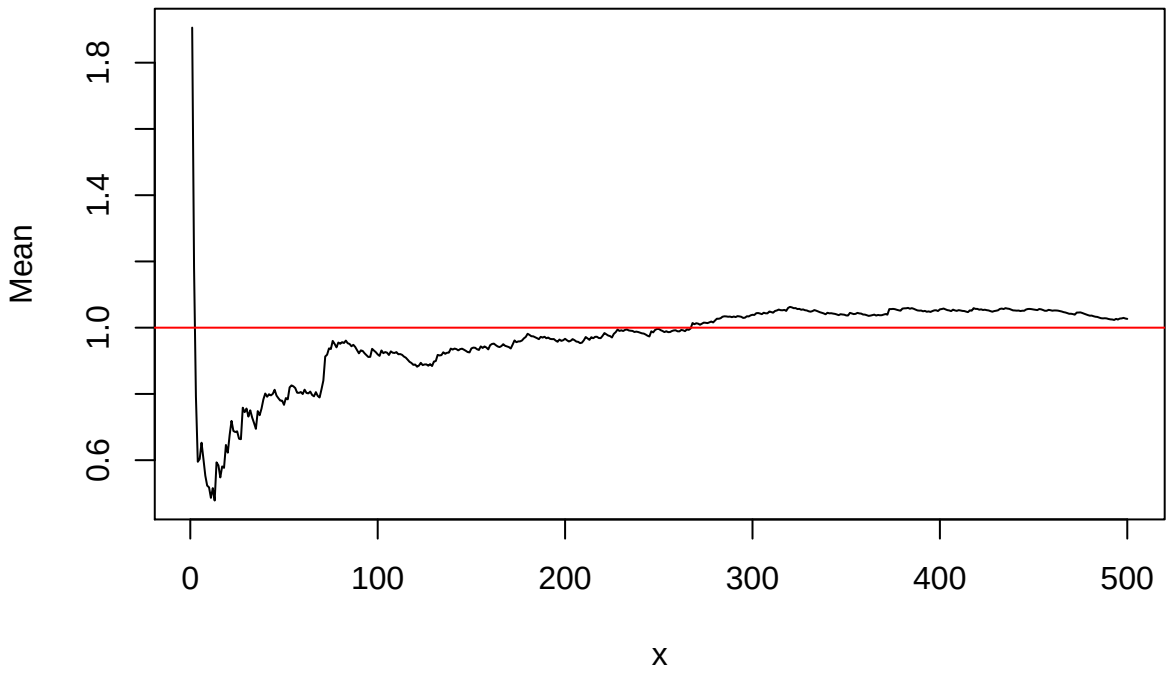
Exercise 1

We generate a N-sample v. Using the recursive relations for $\bar{X}_n$, we derive the function

```
recursivemean<-function(n){
  if(n==0){return(0)}
  else{
    tmp <- recursivemean(n-1)
    return( tmp+(v[n]-tmp)/n )
  }
}
```

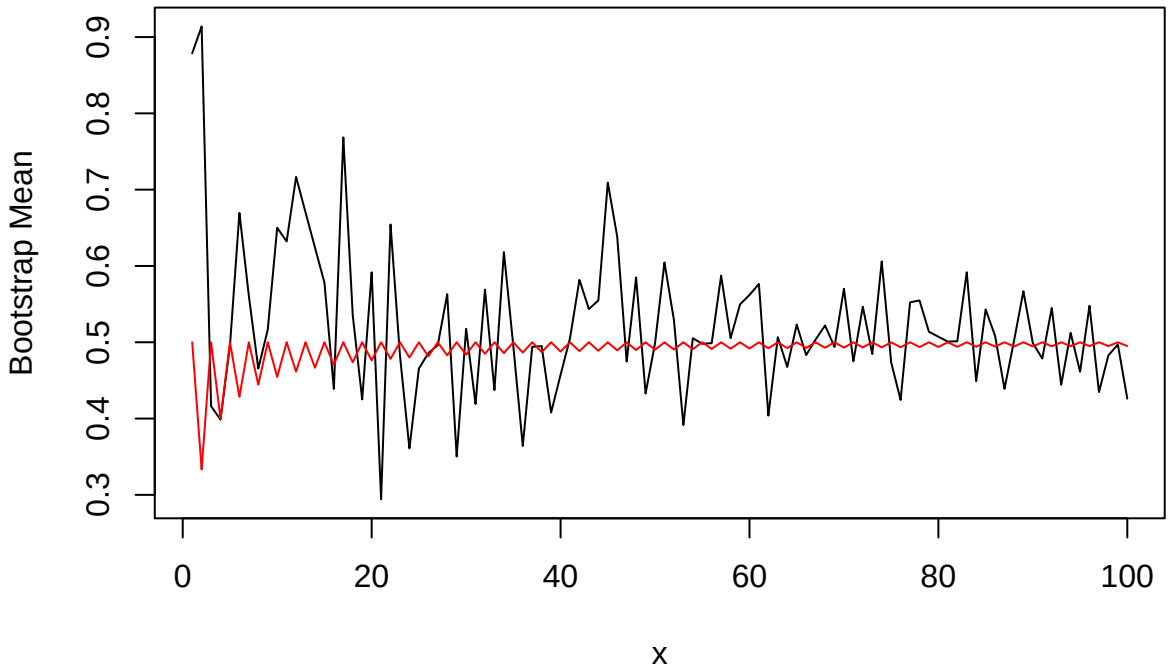
Remark : The variable `tmp` avoids extra recursions in the program and consequently useless time consuming.

```
N <- 500
v <- rexp(N)
x <- 1:N

Mean <- sapply(x,recursivemean)
plot(x,Mean,type='l')
abline(h=1,col='red')
```



Exercise 2

```
bootstrap <- function(n){

    obs <- runif(n)

    mestimate<-median(obs)  # under the empirical distribution, the median of a bootstrap
                            # sample is the median of the observed sample.

    M <- sapply(1:100,function(x) sample(obs,replace=TRUE))
    if(n==1){               # if n=1, M is a vector
      B <- sapply(M,median)
    }else{                  # if n>1, M is a matrix
      B <- apply(M,2,median)
    }

    bootmean <- mean(B)
    bootvar <- mean((B-mestimate)^2)
    return(c(bootmean,bootvar))
}

x <- 1:100
R <- sapply(x,bootstrap)
truemean <- ceiling(x/2)/(x+1)
truevar <- ceiling(x/2)*(x-ceiling(x/2)+1)/((x+1)^2*(x+1))

plot(x,R[1,],type='l',ylab='Bootstrap Mean')
lines(x,truemean,col='red')
```



```r
plot(x,R[2,],type='l',ylab='Bootstrap Var')
lines(x,truevar,col='red')
```