Assignment 2

Harry Woo

2020 5 23

이

library(stats4)

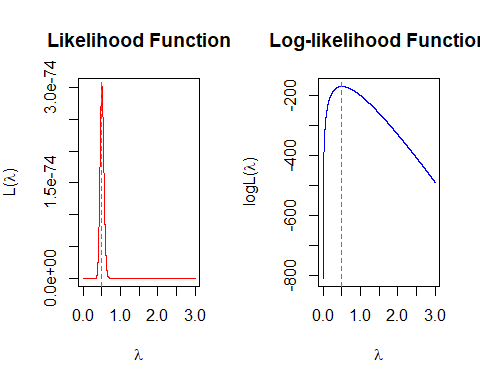
set.seed(12345)  
  
n <- 100  
x <- rexp(n, rate = 1/2)  
  
minuslogL <- function(lambda){  
 -n\*log(lambda) + lambda \* sum(x)  
}  
  
MaxLikeEst <- mle(minuslogL, start = list(lambda = 1/2),  
 method = "L-BFGS-B", lower = 0, upper = Inf)  
  
summary(MaxLikeEst)

## Maximum likelihood estimation  
##   
## Call:  
## mle(minuslogl = minuslogL, start = list(lambda = 1/2), method = "L-BFGS-B",   
## lower = 0, upper = Inf)  
##   
## Coefficients:  
## Estimate Std. Error  
## lambda 0.5001372 0.05001352  
##   
## -2 log L: 338.5748

1/mean(x)

## [1] 0.5001365

xx <- seq(0, 3, length = 10000)  
like <- xx^n \* exp(-xx \* sum(x))  
loglike <- n\*log(xx) - xx \* sum(x)  
  
par(mfrow = c(1,2))  
plot(xx, like, type = "l", xlab = expression(lambda),  
 ylab = expression(paste("L(", lambda, ")")), col = "red",  
 main = "Likelihood Function")  
abline(v = MaxLikeEst@coef, lty = 2, col = "steelblue")  
  
plot(xx, loglike, type = "l", xlab = expression(lambda),  
 ylab = expression(paste(log, "L(", lambda, ")")), col = "blue",  
 main = "Log-likelihood Function")  
abline(v = MaxLikeEst@coef, lty = 2, col = "steelblue")



REP <- 1000  
n <- 20  
mu <- 5  
sigma <- sqrt(2)  
  
sample\_mean <- c()  
sample\_var <- c()  
bias\_mean <- c()  
bias\_var <- c()  
  
for(i in 1:REP) {  
 x = rnorm(n, mean = mu, sd = sigma)  
 sample\_mean[i] <- mean(x)  
 sample\_var[i] <- var(x)  
 bias\_mean[i] <- mu - sample\_mean[i]  
 bias\_var[i] <- sigma^2 - sample\_var[i]  
}  
  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

df <- data.frame(sample\_mean , bias\_mean, sample\_var, bias\_var)  
df %>% summarise\_each(mean)

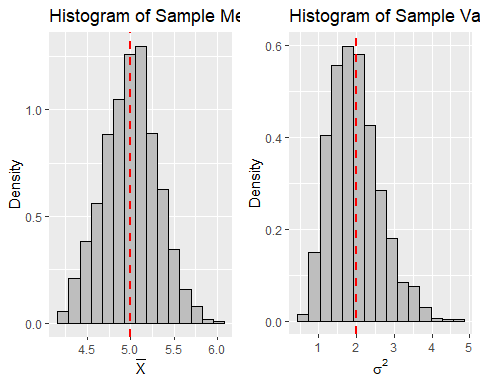
## sample\_mean bias\_mean sample\_var bias\_var  
## 1 4.996784 0.003215907 1.971627 0.02837313

library(ggplot2)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

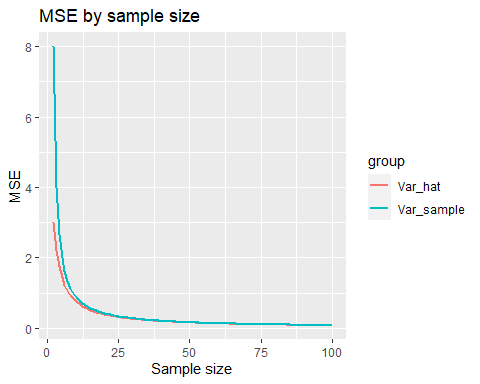
plot1 <- ggplot(data = df, aes(x = sample\_mean, y = ..density..)) +  
 geom\_histogram(fill = "gray", colour = "black", bins = 15) +  
 geom\_vline(xintercept = mu, col = "red", lwd = 1, lty = 2) +  
 xlab(expression(bar(X))) + ylab("Density") +  
 ggtitle(label = "Histogram of Sample Mean")  
   
plot2 <- ggplot(data = df, aes(x = sample\_var, y = ..density..)) +  
 geom\_histogram(fill = "gray", colour = "black", bins = 15) +  
 geom\_vline(xintercept = sigma^2, col = "red", lwd = 1, lty = 2) +  
 xlab(expression(sigma^2)) + ylab("Density") +  
 ggtitle(label = "Histogram of Sample Var")  
  
grid.arrange(plot1, plot2, ncol = 2)



n2 <- seq(2, 100)  
  
mse\_varsam <- (2 \* sigma^4)/(n2 - 1)  
mse\_varhat <- (((2 \* n2) -1)/(n2)^2) \* sigma^4  
  
data.frame(n2, mse\_varsam, mse\_varhat) %>%   
 mutate(compare = mse\_varsam > mse\_varhat) %>%   
 select(compare) %>% table()

## .  
## TRUE   
## 99

df2 <- rbind(data.frame(size = n2, MSE = mse\_varsam, group = "Var\_sample"),  
 data.frame(size = n2, MSE = mse\_varhat, group = "Var\_hat"))  
  
ggplot(data = df2, aes(x = size, y = MSE, colour = group)) +  
 geom\_line(lwd = 1) +  
 xlab("Sample size") + ylab("MSE") +  
 ggtitle(label = "MSE by sample size")



set.seed(123)  
  
n\_100 <- rbinom(100, size = 1, prob = 0.8)  
n\_1000 <- rbinom(1000, size = 1, prob = 0.8)  
n\_10000 <- rbinom(10000, size = 1, prob = 0.8)  
  
mlogL <- function(p,x){  
 -sum(dbinom(x, size = 1, prob = p, log = TRUE))  
}  
  
r1 <- optimize(mlogL, interval = c(0,1), x = n\_100)  
r2 <- optimize(mlogL, interval = c(0,1), x = n\_1000)  
r3 <- optimize(mlogL, interval = c(0,1), x = n\_10000)  
  
data.frame(n\_size = c(100, 1000, 10000),  
 MLE\_by\_R = c(r1$minimum, r2$minimum, r3$minimum),  
 mean = c(mean(n\_100), mean(n\_1000), mean(n\_10000))) %>%   
 mutate(diff = MLE\_by\_R - mean)

## n\_size MLE\_by\_R mean diff  
## 1 100 0.8199952 0.8200 -4.756393e-06  
## 2 1000 0.8050191 0.8050 1.908010e-05  
## 3 10000 0.8048193 0.8048 1.933733e-05