Question 3, Homework 1

Joe Froelicher

9/10/2020

# Part A

set.seed(8675309)  
height <- rnorm(100, 70, sqrt(15))  
head(height)

## [1] 66.14025 72.79561 67.60956 77.85980 74.12634 73.82349

bias <- median(height) - 70  
bias

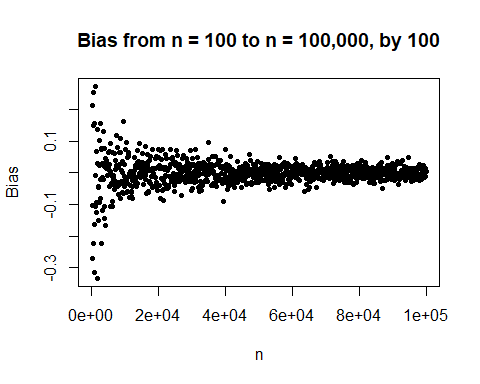
## [1] 0.2121084

# Part B

set.seed(8675309)  
vec.length <- 100000/100  
bias.mat <- matrix(NA, vec.length, 4)  
iter <- 0  
mean <- 70  
sd <- sqrt(15)  
  
for ( i in seq(100, 100000, 100) ) {  
 iter <- iter + 1  
   
 height <- rnorm(i, mean, sd)  
 median <- median(height)  
   
 median.vec <- rep(median, i)  
   
 bias.mat[iter, 1] <- i / 100  
 bias <- median - mean  
 bias.mat[iter, 2] <- bias  
 sum <- 0  
   
 for ( h in height ) {  
 sum <- sum + ( ( h - median ) \*\* 2 )  
 }  
   
 bias.mat[iter, 3] <- sum / length(height)  
 bias.mat[iter, 4] <- iter \* 100  
}  
  
head(bias.mat)

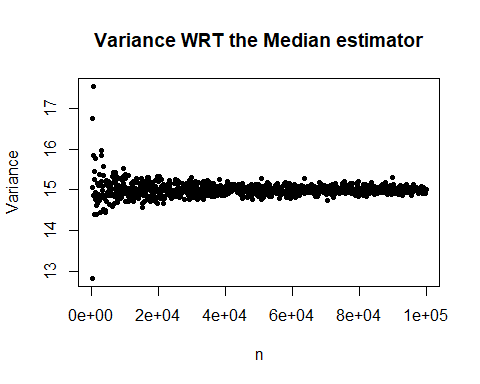
## [,1] [,2] [,3] [,4]  
## [1,] 1 0.2121084 12.82022 100  
## [2,] 2 -0.1025195 15.04912 200  
## [3,] 3 -0.2695723 16.76058 300  
## [4,] 4 0.1501639 15.84363 400  
## [5,] 5 0.2554245 14.85921 500  
## [6,] 6 -0.2223775 17.55084 600

plot(  
 bias.mat[,4],  
 bias.mat[,2],  
 main = "Bias from n = 100 to n = 100,000, by 100",  
 xlab = "n",  
 ylab = "Bias",  
 pch = 20  
)



# Part C

plot(  
 bias.mat[,4],  
 bias.mat[,3],  
 main = "Variance WRT the Median estimator",  
 xlab = "n",  
 ylab = "Variance",  
 pch = 20  
)



# Part D

set.seed(8675309)  
efficiency.mat <- matrix(NA, 10000, 5)  
  
for ( i in seq(1, 10000, 1) ) {  
 height <- rnorm(1000, mean, sd)  
   
 efficiency.mat[i, 1] <- mean(height)  
 efficiency.mat[i, 3] <- median(height)  
   
 mean.var <- 0  
 median.var <- 0  
   
 mean.e <- mean(height)  
 median.e <- median(height)  
   
 for ( h in height ) {  
 mean.var <- mean.var + ( ( h - mean.e ) \*\* 2 )  
 median.var <- median.var + ( ( h - median.e ) \*\* 2 )  
 }  
   
 mean.var <- mean.var / length(height)  
 median.var <- median.var / length(height)  
  
 efficiency.mat[i, 2] <- mean.var  
 efficiency.mat[i, 4] <- median.var  
 efficiency.mat[i, 5] <- i  
}  
  
plot(  
 efficiency.mat[,5],  
 efficiency.mat[,2],  
 Xlab = "n",  
 ylab = "mean variance"  
)

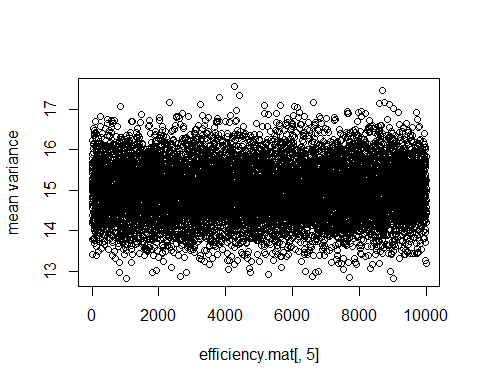
## Warning in plot.window(...): "Xlab" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "Xlab" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "Xlab" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "Xlab" is not a  
## graphical parameter

## Warning in box(...): "Xlab" is not a graphical parameter

## Warning in title(...): "Xlab" is not a graphical parameter



plot(  
 efficiency.mat[,5],  
 efficiency.mat[,4],  
 Xlab = "n",  
 ylab = "median variance"  
)

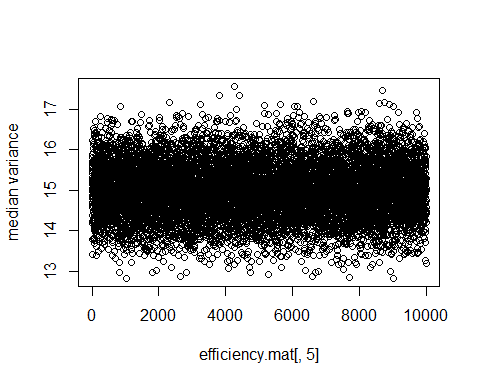
## Warning in plot.window(...): "Xlab" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "Xlab" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "Xlab" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "Xlab" is not a  
## graphical parameter

## Warning in box(...): "Xlab" is not a graphical parameter

## Warning in title(...): "Xlab" is not a graphical parameter



mean(efficiency.mat[,2])

## [1] 14.98337

mean(efficiency.mat[,4])

## [1] 14.9918

In terms of the better estimator, because the variance of the median is closer in magnitude to the assumed 15, then we would call the median a more efficient estimator. However, as mentioned in the next problem, we have a better way of measuring this.

# Part E

The Cramer-Rao lower bound is a function that calculates the minimum variance of an ‘unbiased’ estimator. We know that for an estimator to be efficient, we want the variance of to be greater than or equal to one over the Cramer-Rao function.