HW6\_490IDS\_17

17

10/18/2016

# HW 6 - Due Tuesday October 18, 2016 in moodle and hardcopy in class.

# Upload R file to Moodle with name: HW6\_490IDS\_YourClassID.R

# Do not remove any of the comments. These are marked by

# Please ensure that no identifying information (other than your class ID)

# is on your paper copy, including your name

# We will use the bootstrap technique to generate confidence intervals

# 1. Suppose we have a sample of data from an exponential distribution

# with parameter lambda. In this case use lambda.hat = 1/mean(X).

# As the number of observations increases, does the estimate for lambda

# become roughly normally distributed? We will answer this question in

# the following parts.

# 1a. (1) Generate 100 observations of test data, with lambda=3. Remember

# to set your seed before carrying out any computations.

set.seed(1)  
obs = rexp(n = 100, rate = 3)

# 1b. (1) What is the mean of your test data? (give the code and the value)

mean(obs)

## [1] 0.3435588

# 1c. (1) What is your estimate lambda.hat? (give the code and the value)

lambda.hat = 1/mean(obs)  
print (lambda.hat)

## [1] 2.91071

# 2. Now use the bootstrap to estimate the distribution of

# lambda.hat and create bootstrap confidence intervals for lambda,

# rather than the approach in 1).

# 2a. (1) Form a set of bootstrap estimates of our parameter by generating B

# random samples as you did once in 1a but use lambda.hat since we do not

# know the true lambda in this case (keep n=100). Set B=1000, and again set

# your seed.

set.seed(0)  
library(boot)  
samplemean <- function(x,d) {  
 return(1/mean(x[d]))  
}  
x = rexp(n = 100, rate = lambda.hat)  
bootbet = boot(x,samplemean, 1000)

# 2b. (1) Get a new estimate for lambda.hat from each of the bootstrap samples

# in 2a. You'll want to create a matrix to receive each value. You should

# have 1000 estimates for lambda.hat now.

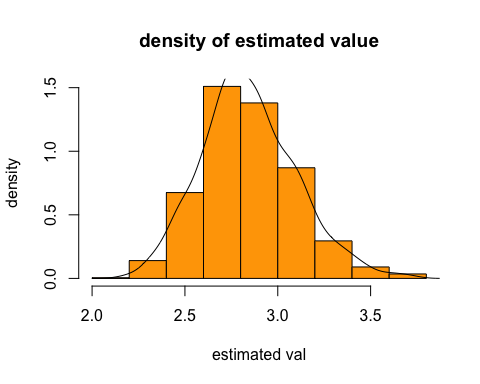
m=as.numeric(bootbet$t)

# 2c. (2) Now look at the sampling distribution for lambda.hat, using the hist

# function. Remember the graphing techniques discussed in class and use them

# to make the plot look professional. Does the distribution look normal?

hist(m,main="density of estimated value", prob = TRUE,xlab = "estimated val",ylab= "density", col = "orange")  
lines(density(m))



print("Yes,this distribution looks normal. It is bell shaped and most of the values are around lambda.hat.")

## [1] "Yes,this distribution looks normal. It is bell shaped and most of the values are around lambda.hat."

# 2d. (1) Calculate an estimate of the standard error of lambda.hat using your

# collection of bootstrap estimated parameters. What is your 95% confidence interval?

std = sd(bootbet$t)  
se = std/sqrt(1000)  
mid = mean(bootbet$t)  
low = mid-1.96\*se  
hi = mid+1.96\*se  
paste("The 95% ci is:", low,hi)

## [1] "The 95% ci is: 2.82736316185876 2.85909301483569"

# 3a. (5) We made some decisions when we used the bootstrap above that we can now question.

# Repeat the above creation of a confidence interval for a range of values of data

# (we had our sample size fixed at 100) and a range of bootstrap values (we had B

# fixed at 1000). Suppose the sample size varies (100, 200, 300, .... , 1000) and

# B varies (1000, 2000, ... , 10000). You will likely find it useful to write

# functions to carry out these calculations. Your final output should be

# upper and lower pairs for the confidence intervals produced using the bootstrap

# method for each value of sample size and B.

# generalize 2b into a function, and vary inputs of sample size and B as we did above.

boot.sample <- function(sample.size, B){  
   
 #code here  
 set.seed(0)  
 library(boot)  
 samplemean <- function(x,d) {  
 return(1/mean(x[d]))  
 }  
 x = rexp(n = sample.size, rate = lambda.hat)  
 bootbet = boot(x,samplemean, B)  
 std = sd(bootbet$t)  
 se = std/sqrt(B)  
 mid = mean(bootbet$t)  
 low = mid-1.96\*se  
 hi = mid+1.96\*se  
 return (c(low,hi))  
}  
a = boot.sample(100,1000)  
b = a[1]  
c = a[2]  
ssize = as.numeric(seq(from = 100, to = 1000, by = 100))  
blist = as.numeric(seq(from = 1000, to = 10000, by = 1000))  
resultl = list()  
resulth = list()  
print("use 100 as sample size")

## [1] "use 100 as sample size"

for(bb in blist){  
 a = boot.sample(100,bb)  
 l = a[1]  
 h = a[2]  
 resultl = c(resultl,l)  
 resulth = c(resulth,h)  
}  
print("use 5000 as b size")

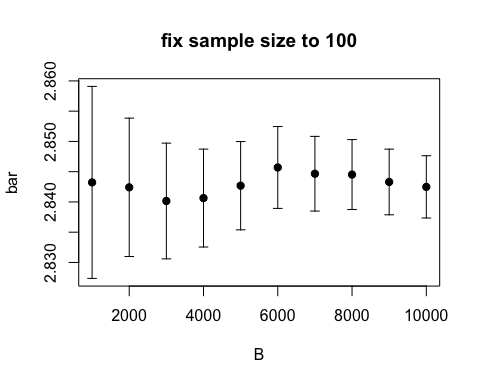
## [1] "use 5000 as b size"

result2l = list()  
result2h = list()  
for(ss in ssize){  
 a = boot.sample(ss,5000)  
 l = a[1]  
 h = a[2]  
 result2l = c(result2l,l)  
 result2h = c(result2h,h)  
}

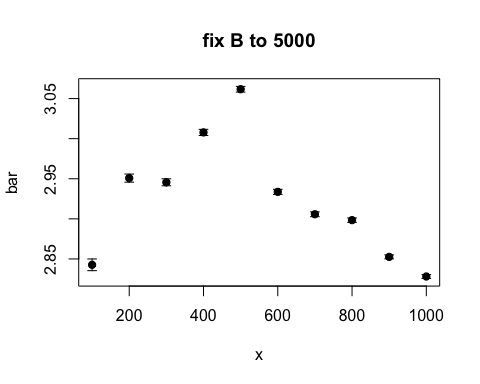
# 3b. (2) Plot your CI limits to show the effect of changing the sample size and

# changing the number of bootstrap replications. What do you conclude?

mid1 = as.numeric(mapply("+",resulth,resultl)/2)  
plot(blist, mid1,ylim=range(resultl, resulth),pch=19, xlab="B", ylab="bar",main="fix sample size to 100")  
arrows(blist,as.numeric(resultl), blist, as.numeric(resulth), length=0.05, angle=90, code=3)



mid2 = as.numeric(mapply("+",result2h,result2l)/2)  
plot(ssize, mid2,ylim=range(result2l, result2h),pch=19, xlab="x", ylab="bar",main="fix B to 5000")  
arrows(ssize,as.numeric(result2l), ssize, as.numeric(result2h), length=0.05, angle=90, code=3)



print("change bootstrap replications makes more sense because the error bound goes smaller when increasing the bootstrap rep size. When fixing B and changing samplesize, it doesn't maek a difference")

## [1] "change bootstrap replications makes more sense because the error bound goes smaller when increasing the bootstrap rep size. When fixing B and changing samplesize, it doesn't maek a difference"

# 4a. (5) In 1961 John Tukey wrote an article called The Future of Data Analysis

# (it is uploaded in moodle). Some people say it is prophetic regarding the

# field of Data Science today. Do you agree or disagee? Why or why not? (Please

# keep your answer less than 500 words).

## [1] "This paper has some impressive insights into the study of data science. The author suggest that data science will be prosper in the future. And it is true. He also suggests that the spread of computer will be useful in data science; this is also true. He correctly anticipated that computer will be very useful when the data set is huge. \nToday, most big firms has high expectation on making use of data, and computer calculation just makes everything convinient. I agree to him most that empirical sampling is important. Although mathematicians may look forward to a ideal case or establishing some sort of theory. Data science is more practical and models must fit the empirical expirience.\nAbstract theory might be good to be treated as a reference, but the model derived must be able to used to real world applications. For example when we are trying to do classification, we might want to be practical about the result, instead of what technique we are using. And in order to make the application work, pure theoratical model might fail.\nLots of experiments needed for optimize the answer and adapt to real world situations.\nI think for some technical detailed he didn't make the prediction very correctly, since there are just so many new technology developped and people are viewing data science in a new angle thanks to these new theories. For example, Bayesian theory is quite useful nowadays, but it is underrated in Tukey's paper.\n "

# 4b. (5) Relate the article to the Life Cycle of Data discussion from class.

# You may wish to choose an example or idea from the article and clearly explore how it

# relates to the Life Cycle of Data. (Please keep your answer less than 500 words).

## [1] "I will briefly discuss the Data processing and analysis part in the Life Cycle of Data. In class we have discussed “every meaningful performance goal which had classically been addressed by mathematical assumptions and theoretical derivations, would in the future be addressed using empirical data and heavy computation.” I will briefly express my thinking toward this.\n In general I think this is very true. 20 years ago if we are looking a question, for example, what are the cause of diabetes. We might select a few parameters that have been reported to cause a higher possibility of diabetes; take them as hypothesis and collect some statistic datas like mean, median, variance to determine if they are really related to diabetes or not.\n Now, thanks to the computing power we have, we can expand our search to more aspects. Before we probably rely on datas from hospital, but now we have hundreds times more dimensions to look into. \n For example, BBC used to publish an article that having fat friend will lead people to become fat. I won't discuss the academic correctness of this article, but it actually gives a new perspective to look into the cause of obesity. The combination of social science, biology and statistics brings the health analysis to a whole new level.\n Back to the question of filtering the cause of diabetes. Classical theory would fail on such tasks because its lack of practical value. There are so many aspects to look into and each will take years to accomplish. How to make it automatic? Can we simply appy theory to every possible field and wait for a hundred years for the solution?\n The answer is no. And that's why empirical data and heavy computation can save us from manually looking for solutions. We can just assume that every cause is possible and have indefinite data input. Design a system that can train itself and calculate the weight to every cause. This whole process will be automatic, which also provides generality.\n If we can grant generality to all type of datas and simply apply computations on them, the analysis of data will be so much easier than make mathematical assumptions and derivations. This is how I think modern day's hypothesis making and evaluation should be."