Homework - Bootstrapping to find best fitting linear regression

Tesfahun Tegene Boshe

12/17/2020

## Introduction

A sample of data with two variables x,y was given. It is known that there is a natural true relationship between those two variables, however, hidden from us because there is some randomness. The task was to find this true relationship using only sample data.

Assuming the relationship to be linear Y = alpha + beta\*x + error

I used bootstrapping to find the best approximations of alpha, beta which minimize error term ‘error’. Since I do not have the who data, I won’t be able to know the error ( I believe the professor will evaluate my solution by comparing my result to the true line.)

boot() - boot() function from boot R package was used for creating samples. lm() - lm() function from stats package was used to fit linear models. pam()- PAM clustering (with 1 cluster center) was used to find the mediods of x,y variables separately.

# load required packages  
requiredPackages = c( "broom","triangle", "readr", "RColorBrewer","boot","cluster") # list of required packages  
for(i in requiredPackages){if(!require(i,character.only = TRUE)) install.packages(i)}

## Loading required package: broom

## Loading required package: triangle

## Loading required package: readr

## Loading required package: RColorBrewer

## Loading required package: boot

## Loading required package: cluster

for(i in requiredPackages){if(!require(i,character.only = TRUE)) library(i,character.only = TRUE) }

## Load the sample data

A sample data of 350 samples (from 1000,000) was given.

# load the data  
sample\_all <- dataset <- read\_delim("C:\\Users\\tesfa\\OneDrive\\Documents\\CoursesDataScince\\Applied Microeconomics\\Materials of Topic 3.2-20201111\\Topic\_3\_2\_MC\_2020\\Topic\_3\_2\_MC\_2020\\sample\_all.csv", ";", escape\_double = FALSE, trim\_ws = TRUE) # load the sample

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## x = col\_double(),  
## y = col\_double()  
## )

## function performing regression

return the coefficients of regression .

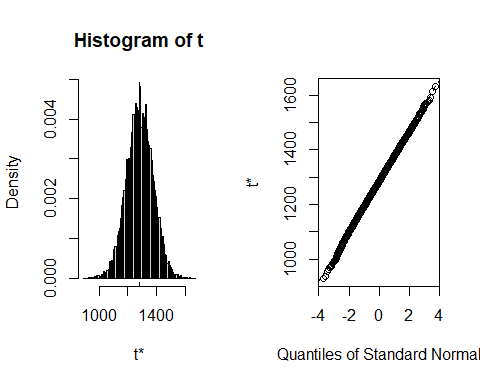
# function to obtain regression weights  
bs <- function(formula, data, indices) {  
 d <- data[indices,] # allows boot to select sample  
 fit <- lm(formula, data=d)  
 return(coef(fit))  
}

## bootstrapping with 10000 replications

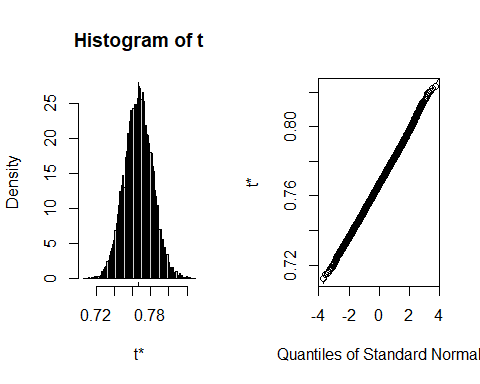
# bootstrapping with 1000 replications  
#   
results <- boot(data=sample\_all, statistic=bs,  
 R=10000, formula=y~x)

## results of bootstrapping

# view results  
plot(results, index=1,, main="Histogram of y - intercepts", col="aquamarine3", breaks=50, probability=TRUE) # y-intercept



# view results  
plot(results, index=2,, main="Histogram of slope values", col="aquamarine3", breaks=50, probability=TRUE) # y-intercept

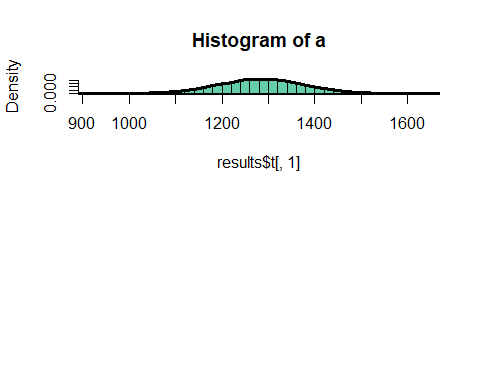


### results for alpha variable

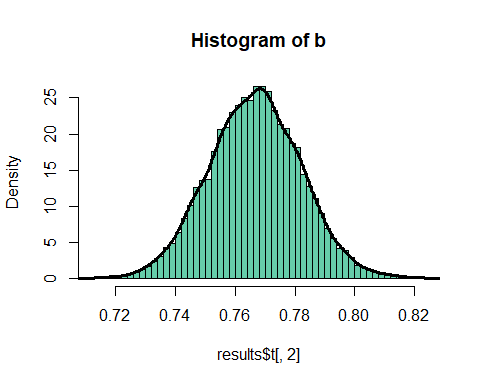
# view results  
par(mfrow=c(2,1)) #to have two graphs in one window one above the other   
#histogram for alfa\_a  
hist(results$t[,1], main="Histogram of a", col="aquamarine3", breaks=50, probability=TRUE)  
lines(density(results$t[,1]), lwd=3)  
#summary of coefficient alfa\_a  
summary(results$t[,1])

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 929 1223 1285 1284 1347 1633

#histogram for alfa\_b

 ### results for beta variable

hist(results$t[,2], main="Histogram of b", col="aquamarine3", breaks=50, probability=TRUE)  
lines(density(results$t[,2]), lwd=3)



#summary of coefficient alfa\_b  
summary(results$t[,2])

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.7124 0.7562 0.7668 0.7667 0.7770 0.8233

par(mfrow=c(1,1)) #return to the screen with one graph displayed

## Clustering

* I clustered the y - intercept and the slope values to find the best measure of center
* PAM clustering was used ( with k = 1 as number of clusters)
* Medoids are the most central elements of the data

#use clustering to find the best approximation of alfa\_median, beta\_edian  
alfa\_a = pam(results$t[,1],1) #cluster::pam(), works for n<65536  
beta\_b = pam(results$t[,2],1) #cluster::pam(), works for n<65536  
alfa\_median = (alfa\_a$medoids[,1])   
beta\_median = (beta\_b$medoids[,1])

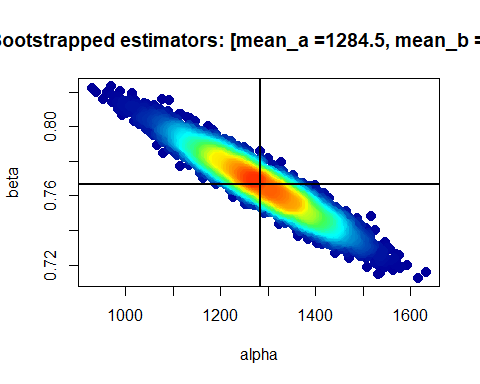
### color plot of points distribution

* Define a function for plotting color by density

# distribution of a and b as XY plot  
#creating a function that will help to make a graph (scatterplot of x and y estimates of 1000 repetitions)  
#densCols produces a vector containing colors which encode the local densities at each point in a scatterplot.  
plot\_colorByDensity = function(x1,x2,  
 ylim=c(min(x2),max(x2)),  
 xlim=c(min(x1),max(x1)),  
 xlab="",ylab="",main="") {  
 df <- data.frame(x1,x2)  
 x <- densCols(x1,x2, colramp=colorRampPalette(c("black", "white")))  
 df$dens <- col2rgb(x)[1,] + 1L  
 cols <- colorRampPalette(c("#000099", "#00FEFF", "#45FE4F","#FCFF00", "#FF9400", "#FF3100"))(256)  
 df$col <- cols[df$dens]  
 plot(x2~x1, data=df[order(df$dens),],   
 ylim=ylim,xlim=xlim,pch=20,col=col,  
 cex=2,xlab=xlab,ylab=ylab,  
 main=main)}  
############################### function end ###################################

# Plot the color by density for the sample data

#using the above function on our data  
plot\_colorByDensity(results$t[,1],results$t[,2],xlab="alpha",ylab="beta",  
 main = paste0("Bootstrapped estimators: [mean\_a =", round( mean(results$t[,1]),1), ", mean\_b =", round( mean(results$t[,2]),2), "]"))  
# abline(h = beta\_b,v = alfa\_b,lwd=2)  
abline(h=mean(results$t[,2]), v=mean(results$t[,1]), lwd=2)



print("alpha = ",); mean(results$t[,1])

## [1] "alpha = "

## [1] 1284.466

print("beta = ",); mean(results$t[,2])

## [1] "beta = "

## [1] 0.7666583

## Comments

* The sampling method used in Bootstrap 2020 Student.R was replaced by boot() function as recommended.
* PAM clustering with 1 center returns the medoid of the data. SInce the data is almost perfectly gaussian, I believe the mean can be a better measure of center.
* boot package does a much faster sampling.

## References

[1] AM 2020 Topic 3.2 Bootstrapping ok at moodle page for the course ‘Advanced Microeconomics’ at UW.

[2] <https://www.statmethods.net/advstats/bootstrapping.html>

[3] Sample code ‘Bootstrap 2020 Student’ at moodle page for the course ‘Advanced Microeconomics’ at UW.