Bootstrapping on the Liver Steatosis dataset variables Cholesterol and Diabetes for Morbidly Obese Patients

Kimberly Schveder

Spring 2020 ~ Advanced Statistical Computing with Dr. Ye

3/16/20

1. Introduction to Bootstrapping Project.

The Liver Steatosis dataset from the Clinic Clinic's data bank was used in this analysis. Bootstrapping was used to get bootstrapped estimates of the 400+ patients being tested for liver steatosis at the Cleveland Clinic. The variables cholesterol and diabetes status were of main curiosity here. The population mean Cholesterol level in patients at risk of having or having Liver Steatosis was estimated with a bootstrapping random sample of 10000. A hypothesis test of Diabetes status of the patients on cholesterol was also completed via bootstrapping random sampling. The sample size for the hypothesis test was 5000.

2. Explanation of the Liver Steatosis dataset.

The liver steatosis data has 443 rows of patient data. Each of the patients are morbidly obese and are being tested for liver steatosis and fibrosis. The cholesterol and diabetes were the chosen variables to analyze and answers the research questions. The questions here are what is the mean cholesterol level and is there any difference in the mean cholesterol levels between the patients with diabetes and without diabetes?

3. Details of the methods used in the analysis.

Bootstrap random sampling works by taking samples with replacement of a certain size from a dataset in order to mimic what the population data would look like. According to the Singe et al,

> "The purpose of a sample study is to gather information cheaply in a timely fashion. The idea behind bootstrap is to use the data of a sample study at hand as a 'surrogate population', for the purpose of approximating the sampling distribution of a statistic; i.e. to resample (with replacement) from the sample data at hand and create a large number of 'phantom samples' known as bootstrap samples" (2).

A summary sample statistic $\hat{\theta}$ is computed from such data to estimate the population parameter θ. Such bootstrap data is typically displayed in a density distribution or a histogram to show the normality (or

sometimes even lack of normality) of the bootstrapped data. The bias and standard error are also typically calculated for such a summary sample statistic.

The bootstrapping was done with the boot package in R. The functions used from this package were the boot() and the boot.ci() functions. The boot function finds the bootstrapping statistics given a bootstrap sample size (that is larger than the dataset size) and the boot.ci function find the bootstrap CI for the parameter, given a significance level and bootstrap sample size. The statistic that is used is specified by an external function that is defined by the user that takes the data and finds the statistic, such as the mean or the two-sample t-test mean difference to find the p-value. Both of these statistic functions are defined in the R script.

4. Results of the data analysis.

The bootstrapped mean of the cholesterol level from a bootstrap sample of size 10000 is about 183.87 mg/dL, a relatively high cholesterol level. The bias for this estimated mean is relatively small, 0.0232, and the standard error is 1.95, according to Figure 1. The bootstrap confidence interval calculations are also based on the 10000 bootstrap sample size. The 95% confidence interval for the true mean cholesterol level is [180.2, 187.8]. We are 95% confident that the true mean cholesterol level of all people at risk of or having liver steatosis or fibrosis is between 180.2 and 187.8 mg/dL.

```
> # create the mean function given a set of data
> # (a data frame for example)
> # and a row number i
> meanfun <- function(data, i){
+   #selecting sample with the boot function below
+   d <- data[i, ]
+   return(mean(d))
+ }
> #use the boot function on the xs column
> #in the data, using the meanfun
> #function created above with a bootstrap random sample of 10000
> bo <- boot(CHOL[, "CHOL", drop = FALSE], statistic=meanfun, R=10000)
> bo

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = CHOL[, "CHOL", drop = FALSE], statistic = meanfun,
    R = 10000)


Bootstrap Statistics :
    original     bias    std. error
t1* 183.8781 0.02318194    1.951225
> boot.ci(bo, conf=0.95, type="bca")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates

CALL :
boot.ci(boot.out = bo, conf = 0.95, type = "bca")

Intervals :
Level       BCa
95%   (180.2, 187.8 )
Calculations and Intervals on Original Scale
> plot(bo)
```

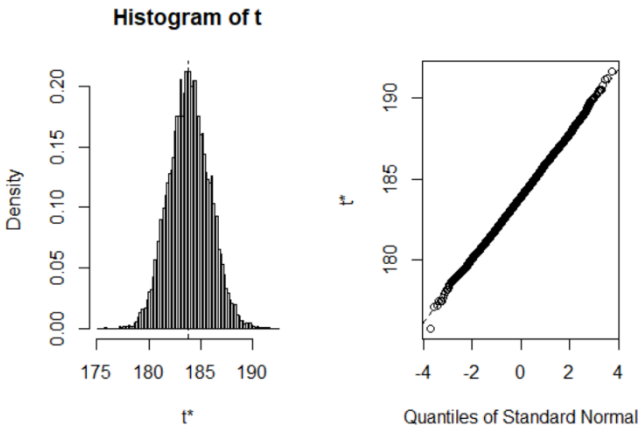

Figure 1: the code and output for the bootstrap mean.

Notice that the density plot of the bootstrap sample t* is normally distributed. This observation is also confirmed with the QQ plot in Figure 1, where all of the data points line-up nearly perfectly along the line.

Next, the bootstrapped test of the difference in the mean cholesterol levels between people who have diabetes and who do not have diabetes, but who are all at risk of or have liver steatosis or fibrosis was found, as shown in Figure 2 below. The hypothesis test is denoted as follows: the null hypothesis is H_0: $\mu_{Diabetes} = \mu_{No\ Diabetes}$ vs. the alternative hypothesis, which is H_0: $\mu_{Diabetes} \neq \mu_{No\ Diabetes}$. In Figure 2 below, the bootstrap sample of size 5000 was found and the mean different was found to be -1.02019, with a bias of 0.969 and standard error of 3.957369, according to Figure 2. The p-value for the test is 0.7968, which is greater than 5% or any other reasonable p-value. Therefore, the null hypothesis is not rejected and it is safe to assume that there is no apparent difference in the mean cholesterol levels between people who have diabetes and who do not have diabetes, but who are all at risk of or have liver steatosis or fibrosis was found.

```
> CHOL_by_DM <- data.frame(CHOL,DM)
> require(boot)
> set.seed(1234)
> diff2 = function(d1,i){
+   d = d1;
+   d$DM <- d$DM[i];  # randomly re-assign groups
+   Mean= tapply(X=d$CHOL, INDEX=d$DM, mean)
+   Diff = Mean[1]-Mean[2]
+   Diff
+ }
> set.seed(1234)
> b4 = boot(data = CHOL_by_DM, statistic = diff2, R = 5000)
> b4

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = CHOL_by_DM, statistic = diff2, R = 5000)


Bootstrap Statistics :
    original    bias    std. error
t1* -1.02019 0.9693123    3.957369
> mean(abs(b4$t) > abs(b4$t0)) #the p-value
[1] 0.7968
```

Figure 2: Bootstrap output for the bootstrap hypothesis test.

5. Discussions and conclusions.

There are a couple of main take-aways here. People who are at risk of having liver steatosis or fibrosis are likely to have a cholesterol level of about 184 mg/dL. Perhaps, morbidly obese patients with such a cholesterol level should be tested for such diseases. This should be taken into consideration even if the patients have diabetes or not.

6. References and computer code.

The dataset represents data from the study by Wu et al. "Prevalence of Liver Steatosis and Fibrosis and the Diagnostic Accuracy of Ultrasound in Bariatric Surgery Patients". Obes Surg 2012; 22: 240-247.

Cleveland Clinic dataset bank: https://www.lerner.ccf.org/qhs/datasets/datasets.php

http://www.stat.rutgers.edu/home/mxie/rcpapers/bootstrap.pdf

```
#R code:
#set working directory
setwd("C:/Users/kasch/Dropbox/Statistics Career Stuff/Spring
2020/Advanced Statistical Computing")

#data <- data.frame(xs = rnorm(15, 2))

LiverSteatosisData <- read.csv("LiverSteatosis.csv")

#####The bootstrapped mean#####
CHOL <- LiverSteatosisData[,"CHOL"]
# CHOL <- as.numeric(CHOL)
#impute missing values with the median CHOL value.
CHOL[which(is.na(CHOL))] <- median(CHOL, na.rm = T)
#create dataframe out of revised CHOL data
CHOL <- as.data.frame(CHOL)


#use library boot:
# install.packages("boot")
library(boot)

# create the mean function given a set of data
# (a data frame for example)
# and a row number i
meanfun <- function(data, i){
  #selecting sample with the boot function below
  d <- data[i, ]
  return(mean(d))
}
```

```
#meanfun(LiverSteatosisData, 12)

#use the boot function on the xs column
#in the data, using the meanfun
#function created above with a bootstrap random sample of 10000
bo <- boot(CHOL[, "CHOL", drop = FALSE], statistic=meanfun, R=10000)
boot.ci(bo, conf=0.95, type="bca")
plot(bo)

#sources: https://data-flair.training/blogs/bootstrapping-in-r/

######The boostraped ANOVA hypothesis test######

#toy data from
https://stats.stackexchange.com/questions/20701/computing-p-value-
using-bootstrap-with-r

# time = c(14,18,11,13,18,17,21,9,16,17,14,15,
#          12,12,14,13,6,18,14,16,10,7,15,10)
# group=c(rep(1:2, each=12))
# sleep = data.frame(time, group)

DM <- LiverSteatosisData[,"DM"]
#impute the two NA values with a random
#value of DM status sample(DM, 1) to keep things simple
DM[which(is.na(DM))] <- sample(DM, 1)
#in future research, could use ML techinques,
#such as imputation with random forests
# to solve the problem of missing values:
#https://www.researchgate.net/post/What_is_the_proper_imputation_metho
d_for_categorical_missing_value

CHOL_by_DM <- data.frame(CHOL,DM)

require(boot)

diff <- function(d1,i){#,num_vec =
CHOL_by_DM$CHOL,group_vec=CHOL_by_DM$DM){
  #num_vec is the vector of the numbers from the data frame,
    #such as time or cholestrol values.
  #group_vec is the vector of the categories
    # or groups that are being compared,
    #such as people with diabetes vs. people without diabetes.
    d = d1[i,]
    #Mean= tapply(X=d$time, INDEX=d$group, mean)
    Mean= tapply(X=CHOL_by_DM$CHOL, INDEX=CHOL_by_DM$DM, mean)
    Diff = Mean[1]-Mean[2]
    Diff
}

#diff <- diff(CHOL_by_DM, i,CHOL_by_DM$CHOL,CHOL_by_DM$DM)
```

```
#CHOL_by_DM

set.seed(1234)
#b3 = boot(data = sleep, statistic = diff, R = 5000,
strata=sleep$group)
b3 = boot(data = CHOL_by_DM, statistic = diff, R = 5000,
strata=CHOL_by_DM$DM)

quantile(b3$t,c(0.025,0.975))

diff2 = function(d1,i){
  d = d1;
  d$DM <- d$DM[i];  # randomly re-assign groups
  Mean= tapply(X=d$CHOL, INDEX=d$DM, mean)
  Diff = Mean[1]-Mean[2]
  Diff
}

set.seed(1234)
b4 = boot(data = CHOL_by_DM, statistic = diff2, R = 5000)
mean(abs(b4$t) > abs(b4$t0)) #the p-value
```