PROJECT 4

Abstract

. The scope of this project is dealing with Chapter 11 of Bain, L.J. and Engelhardt M. Introduction to Probability and Mathematical Statistics 1992.

Question 3:

1. Problem Statement:

Given a random sample, $X_1, ..., X_n$ with $X_i \sim N(\mu, \sigma^2)\ \forall i = 1, ..., n$ with both population parameters unknown. Design a simulation study to empirically show the behavior of the *Percentile Bootstrap Confidence Intervals* for μ and σ^2 with respect to:

1) Small vs. Large Sample Size
2) Symmetry of the Sampling Distribution

2. Theoretical Background, Methodology and Design:

Theoretical Background: From Project 3 we saw the following theoretical layout of the Percentile Bootstrap Confidence Interval Method: The percentile method, a certain percentage (e.g. 5% or 10%) is trimmed from the lower and upper end of the sample statistic (e.g. the mean or standard deviation). Which number you trim depends on the confidence interval you're looking for. For example, a 90% confidence interval would generate a 100% – 90% = 10% trim (i.e. 5% from both ends) - Glen (2016).

A bootstrap percentile confidence interval of $\widehat{\theta}$,an estimator of θ, can be obtained as follows:

1) Generate B number of random bootstrap samples,
2) Calculate a parameter estimate from each bootstrap sample,
3) Order all B bootstrap parameter estimates from the lowest to highest,
4) Construct the confidence interval: $[\widehat{\theta}_{lower\,limit}, \widehat{\theta}_{upper\,limit}]$=$[\widehat{\theta}^*_j, \widehat{\theta}^*_k]$ such that $\widehat{\theta}^*_j$ denotes the j'th quantile (the lower limit) and $\widehat{\theta}^*_k$ denotes the k'th quantile (the upper limit). Also $j = [\frac{\alpha}{2} \times B]$ and $k = [(1 - \frac{\alpha}{2}) \times B]$. Thus, a 95% percentile bootstrap CI with 1,000 bootstrap samples is the interval between the 25th quantile value and the 975th quantile value of the 1,000 bootstrap parameter estimates - Jung et al. (2019).

Methodology and Design: The goal behind the design of this simulation study is to graphically illustrate the implementation of the Percentile Bootstrap Confidence Interval Method and to compare the results for a small and large sample size. This method will be applied for both μ and σ^2 with each parameter's density obtained from the bootstrap samples plotted on a separate axes to show the difference between a small and large sample size as well as to illuminate differences in symmetry. I will also plot the mean of the different parameter estimates' bootstrap samples in order to see if it falls within the Confidence Interval calculated.

3. Implementation of Simulation Study:

Simulation study for the parameter μ:

```
n_S = 100
n_L = 1000
mu = 3
sigma = 2

set.seed(128)
x_small = rnorm(n_S, mu , sigma)
x_large = rnorm(n_L, mu , sigma)

n_boot = 1000 #number of bootstrap iterations

#Generate Bootstrap Sample
xBootSample_small = replicate(n_boot, sample(x_small, n_S, replace = TRUE))
xBootSample_large = replicate(n_boot, sample(x_large, n_L, replace = TRUE))

#reSampling_Means consists of 1000 mean values each one for every sub sample of size n.
smallSample = replicate(n_boot, mean(sample(x_small,n_S,replace=TRUE)))
largeSample = replicate(n_boot, mean(sample(x_large,n_L,replace=TRUE)))

#Ordering the bootstrap parameter estimates from lowest to highest
ordered_mean_ests_S = smallSample[order(smallSample)]
ordered_mean_ests_L = largeSample[order(largeSample)]

#For 95% Percentile Confidence Interval
quantsU_mean_S = quantile(ordered_mean_ests_S, probs = 0.975)
quantsL_mean_S = quantile(ordered_mean_ests_S, probs = 0.025)
#-------------------------------------
quantsU_mean_L = quantile(ordered_mean_ests_L, probs = 0.975)
quantsL_mean_L = quantile(ordered_mean_ests_L, probs = 0.025)

df_means = data.frame(smallSample, largeSample)
data<- melt(df_means)

ggplot(data, aes(x=value, fill= variable)) + geom_density(alpha=0.4) +
  geom_vline(aes(xintercept=mean(smallSample)),color="red", linetype="dashed", size=1)+
```

```
geom_vline(aes(xintercept=mean(largeSample)),color="blue", linetype="dashed", size=1)+
geom_vline(aes(xintercept=quantsU_mean_S),color="red", linetype="solid", size=0.7)+
geom_vline(aes(xintercept=quantsL_mean_S),color="red", linetype="solid", size=0.7)+
annotate(geom = "text",
         label = c(as.numeric(round(quantsU_mean_S,3)),
                   as.numeric(round(quantsL_mean_S,3))),
         x = c(quantsU_mean_S+0.02, quantsL_mean_S+0.02), y = c(3, 3),
         angle = 90, vjust = 1,color = "red") +
geom_vline(aes(xintercept=quantsU_mean_L), color="blue", linetype="solid", size=0.7)+
geom_vline(aes(xintercept=quantsL_mean_L), color="blue", linetype="solid", size=0.7)+
annotate(geom = "text",
         label = c(as.numeric(round(quantsU_mean_L,3)),
                   as.numeric(round(quantsL_mean_L,3))),
         x = c(quantsU_mean_L+0.02, quantsL_mean_L+0.02), y = c(3, 3),
         angle = 90, vjust = 1, color = "blue") +
annotate(geom = "text",
         label = c(as.numeric(round(mean(largeSample),3)),
                   as.numeric(round(mean(smallSample),3))),
         x = c(mean(largeSample)+0.02, mean(smallSample)+0.02), y = c(3, 3),
         angle = 90, vjust = 1, color = "black") +
labs(title = "Frequency Plot of Bootstrap Parameter Estimates for the Mean",
     x = "Parameter Estimates of the Mean", y = "Frequency",
     caption = "Figure 1",
     color = "Legend") +
scale_color_manual(values = colors) +
theme(plot.title = element_text(hjust = 0.5, face = "bold"),
      plot.caption = element_text(hjust = 0.5, colour = "blue"),
      legend.justification = c("left", "top"),
      legend.position = c(.03, .65),
      legend.box.just = "right",
      legend.margin = margin(6, 6, 6, 6),
      axis.line = element_line(size = 0.5, colour = "black", linetype=1))
```

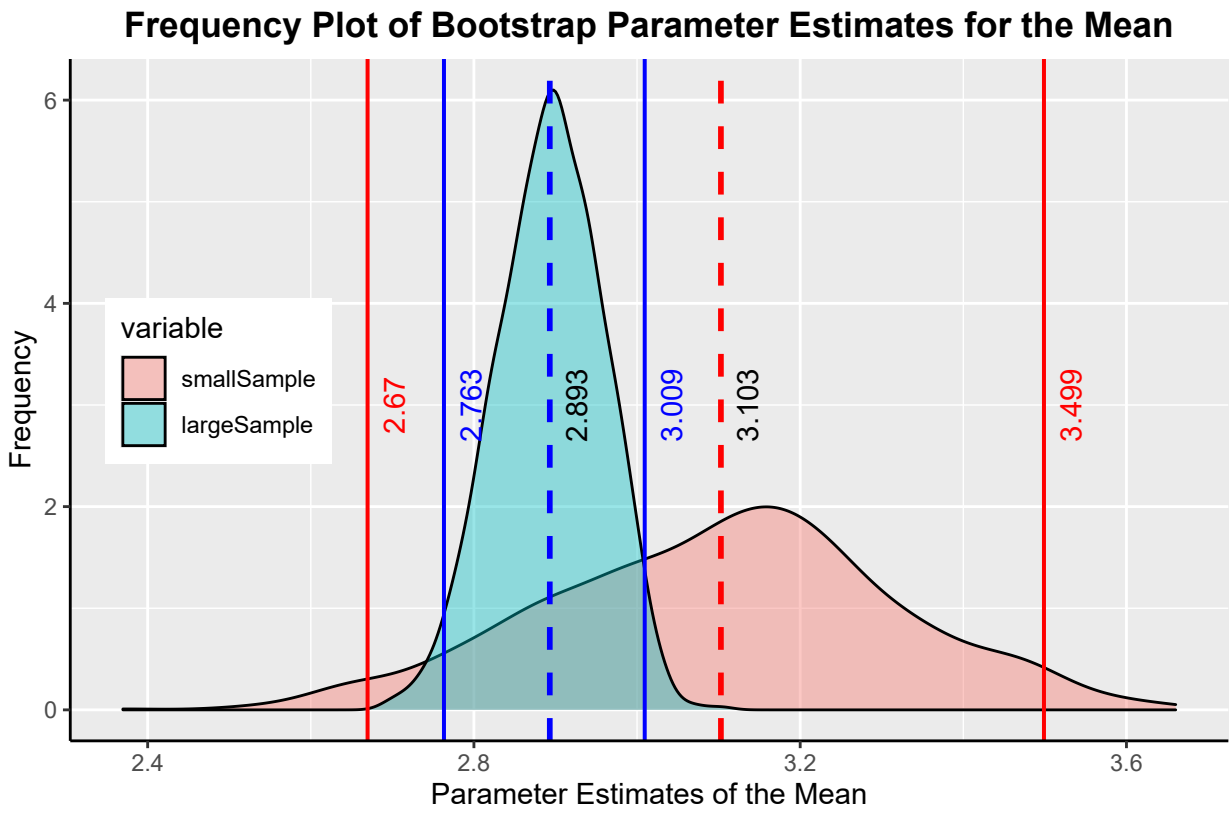

Figure 1

Simulation study for the parameter σ^2:

```
n_S = 100
n_L = 1000
mu = 3
sigma = 2

set.seed(128)
x_small = rnorm(n_S, mu , sigma)
x_large = rnorm(n_L, mu , sigma)

n_boot = 1000 #number of bootstrap iterations

#Generate Bootstrap Sample
xBootSample_small = replicate(n_boot, sample(x_small, n_S, replace = TRUE))
xBootSample_large = replicate(n_boot, sample(x_large, n_L, replace = TRUE))

#reSampling_Var consists of 1000 var values each one for every sub sample of size n.
smallSample = replicate(n_boot, var(sample(x_small,n_S,replace=TRUE)))
largeSample = replicate(n_boot, var(sample(x_large,n_L,replace=TRUE)))

#Ordering the bootstrap parameter estimates from lowest to highest
ordered_var_ests_S = smallSample[order(smallSample)]
ordered_var_ests_L = largeSample[order(largeSample)]
```

```r
#For 95% Percentile Confidence Interval
quantsU_var_S = quantile(ordered_var_ests_S, probs = 0.975)
quantsL_var_S = quantile(ordered_var_ests_S, probs = 0.025)
#-------------------------------------
quantsU_var_L = quantile(ordered_var_ests_L, probs = 0.975)
quantsL_var_L = quantile(ordered_var_ests_L, probs = 0.025)

df_vars = data.frame(smallSample, largeSample)
data<- melt(df_vars)

ggplot(data, aes(x=value, fill= variable)) + geom_density(alpha=0.4)+
  geom_vline(aes(xintercept=mean(smallSample)),color="red", linetype="dashed", size=1)+
  geom_vline(aes(xintercept=mean(largeSample)),color="blue", linetype="dashed", size=1)+
  geom_vline(aes(xintercept=quantsU_var_S),color="red", linetype="solid", size=0.7)+
  geom_vline(aes(xintercept=quantsL_var_S),color="red", linetype="solid", size=0.7)+
  annotate(geom = "text",
           label = c(as.numeric(round(quantsU_var_S,3)),
                     as.numeric(round(quantsL_var_S,3))),
           x = c(quantsU_var_S+0.02, quantsL_var_S+0.02), y = c(1.5, 1.5),
           angle = 90, vjust = 1,color = "red") +
  geom_vline(aes(xintercept=quantsU_var_L),color="blue", linetype="solid", size=0.7)+
  geom_vline(aes(xintercept=quantsL_var_L),color="blue", linetype="solid", size=0.7)+
  annotate(geom = "text",
           label = c(as.numeric(round(quantsU_var_L,3)),
                     as.numeric(round(quantsL_var_L,3))),
           x = c(quantsU_var_L+0.02, quantsL_var_L+0.02), y = c(1.5, 1.5),
           angle = 90, vjust = 1, color = "blue") +
  annotate(geom = "text",
           label = c(as.numeric(round(mean(largeSample),3)),
                     as.numeric(round(mean(smallSample),3))),
           x = c(mean(largeSample)+0.02, mean(smallSample)+0.02), y = c(1.5, 1.5),
           angle = 90, vjust = 1, color = "black") +
  labs(title = "Frequency Plot of Bootstrap Parameter Estimates for the Variance",
       x = "Parameter Estimates of the Variance", y = "Frequency",
       caption = "Figure 2",
       color = "Legend") +
  scale_color_manual(values = colors) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        plot.caption = element_text(hjust = 0.5, colour = "blue"),
        legend.justification = c("left", "top"),
        legend.position = c(.01, .65),
        legend.box.just = "right",
        legend.margin = margin(6, 6, 6, 6),
        axis.line = element_line(size = 0.5, colour = "black", linetype=1))
```

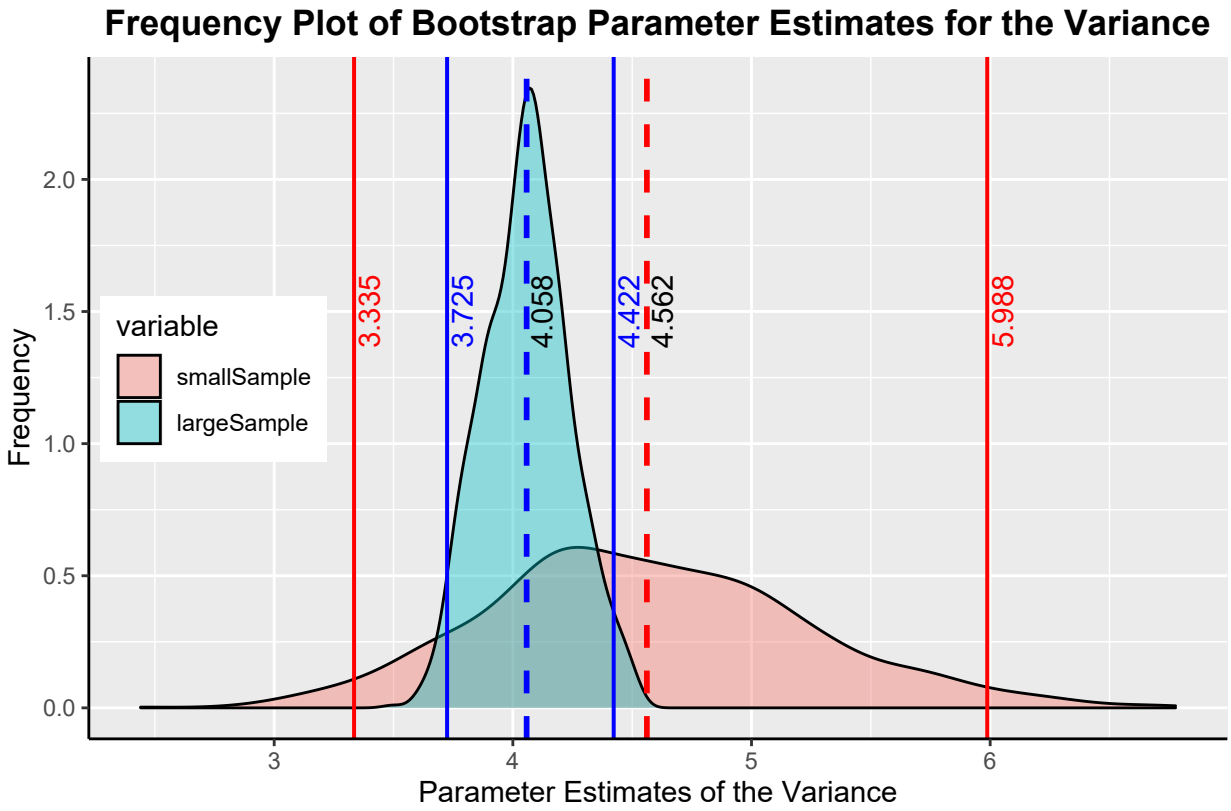

Figure 2

4. Discussion fo Results Obtained:

Firstly, from both Figure 1 and 2 we see that, as we increase the sample size, we get:

1) a decrease in variability of the bootstrap distribution as is evident by the decrease in the span of the 95% confidence interval as indicated by the thin blue lines for a large sample size and the thin red lines for a small sample size.

2) an increase in symmetry around the mean of the bootstrap distribution as seen in both figure 1 and 2 with the mean of the distribution indicated by the dark dotted line.

3) we will get a more precise estimate of the parameter in question as the symmetry and smoothness of the bootstrap sample increases

4) the shape of the bootstrap distribution seems to tend to the shape of the original population sample

5) the more bootstrap samples we use for a given sample size, the more precise our estimate will become

Secondly, we should take note of the following cautions when it comes to the bootstrap method:

1) Using this method to construct confidence intervals only work if the bootstrap distribution is approximately smooth and symmetric

2) When the bootstrap distribution is highly skewed or looks spiky with gaps, we will need to introduce other methods for constructing statistically significant confidence intervals for the parameters in question

Thus, in conclusion we see that the bootstrap method can be used to construct confidence intervals for any desired $100(1-\alpha)\%$, but that caution should be had when analyzing these intervals when there is questions about the smoothness and symmetry of the bootstrap distribution.

References

Glen, Stephanie. 2016. "*Bootstrap Sample: Definition, Example*.". http://www.statisticshowto.com/bootstrap-sample/ [Accessed: 2021/10/27].

Jung, Lee, Gupta and Cho. 2019. "*Comparison of Bootstrap Confidence Interval Methods for GSCA Using a Monte Carlo Simulation*.". http://www.frontiersin.org/articles/10.3389/fpsyg.2019.02215/full#h3 [Accessed: 2021/10/27].