Report Exercise 4 **Ruth Kristianingsih**
Non-Parametric Bootstrap Date: 15 November 2017
Prof. Pere Puig

Problem Statement 1

The following data set corresponds to the feed conversion indexes of 24 calves coming from one lot:

3.65 4.03 4.58 4.61 4.70 4.85 3.21 3.93 3.15 3.00 2.93 3.56 4.13 3.68 3.88 3.25 3.92 3.99 3.04 3.10 3.20 3.35 3.19 3.10

Calculate 95 % confidence interval based on non parametric bootstrap for the population trimmed mean(25 %), standard deviation and coefficient of variation.

Problem 1- Analysis

In animal husbandry, feed conversion ratio (FCR) or feed conversion rate is a ratio or rate measuring of the efficiency with which the bodies of livestock convert animal feed into the desired output. For dairy cows, for example, the output is milk, whereas in animals raised for meat (such as beef cows, pigs, chickens, and fish) the output is the flesh, that is, the body mass gained by the animal, represented either in the final mass of the animal or the mass of the dressed output. FCR is the mass of the input divided by the output (thus mass of feed per mass of milk or meat).

Figure 1: Calves in one lot

In this analysis, the Confidence Interval of Trimmed Mean (25 %), standard error, and coefficient of variation will be calculated using bootstrap method. For each statistics, the three methods will be conducted (classical method, quantile or percentile method, and simple method).

Estimation of Confidence Interval for Trimmed Mean

Firstly, the confidence interval (CI) for the trimmed mean calculation will be calculated.

A truncated mean or trimmed mean is a statistical measure of central tendency, much like the mean and median. It involves the calculation of the mean after discarding given parts of a probability distribution or sample at the high and low end, and typically discarding an equal amount of both. This number of points to be discarded is usually given as a percentage of the total number of points, but may also be given as a fixed number of points.

For most statistical applications, 5 to 25 percent of the ends are discarded; the 25 % trimmed mean (when the lowest 25 % and the highest 25% are discarded) is known as the interquartile mean. For example, given a set of 8 points, trimming by 12.5 % would discard the minimum and maximum value in the sample: the smallest and largest values, and would compute the mean of the remaining 6 points.

Using the R syntax below, the CI of trimmed mean of the population can be calculated.

```r
fci.calves <- c(3.65, 4.03, 4.58, 4.61, 4.70, 4.85, 3.21, 3.93, 3.15, 3.00,
    2.93, 3.56, 4.13, 3.68, 3.88, 3.92, 3.99, 3.04, 3.10, 3.20, 3.35, 3.19,
    3.10)
fci.sort <- sort(fci.calves)
trim.mean <- mean(fci.sort, trim= 0.25)
trim.mean
#=====================================================
n = length(fci.sort)
#set number of bootstrap samples
nsim=1000
stat = numeric(nsim) #create a vector in which to store the results
#Set up a loop to generate a series of bootstrap samples
for (i in 1:nsim){
  fciB = sample (fci.sort, n, replace=T)
  fci.sortB <- sort(fciB)
  stat[i] = mean(fci.sortB, trim=0.25)}
#estimated standard error (classical to calculate CI)
#=====================================================
mean(stat)
sd(stat)
##percentile method
#===================
quan <- quantile(stat,c(0.025,0.975))
boxplot (stat, main= "Boxplot of Feed Conversion Indexes of Calves - Botstrap")
hist(stat, main="Histogram - Precentile/Quantile Method")
abline(v=quan,col="red")
##simple method
#===============
a <- 2*trim.mean-quantile(stat,0.975)
b <- 2*trim.mean-quantile(stat,0.025)
hist(stat, main="Histogram - Simple Method")
abline(v=a,col="red")
abline(v=b,col="red")
```

Figure 2 shows the boxplot of the bootstraping results where there are some outliers of the data.

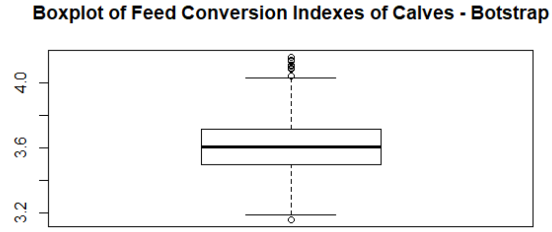

Figure 2: Boxplot of Bootstrap Simulation

Using the classical method to calculate the confidence interval, the results is 3.61 ± 0.32. In comparison of using the quantile or percentile method, the result is 3.62 ± 0.3. The confidence interval can easily be seen in figure 4.

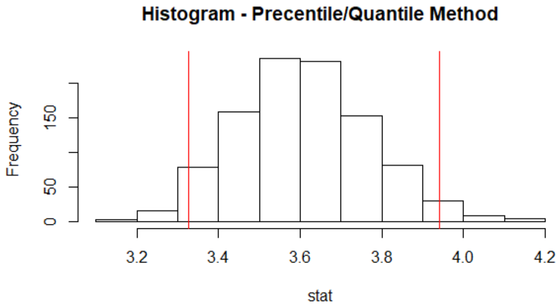

Figure 3: Histogram of Bootstrap Simulation

Using the simple method, the confidence interval can be found which is 3.55 ± 0.3. And the following histogram shows where the confidence interval lies.

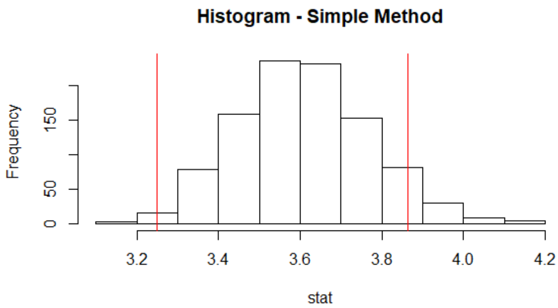

Figure 4: Histogram of Bootstrap Simulation

However, since as it can be seen that the histogram is not very symmetric, simple method will be more accurate to calculate confidence interval using simple method which has different approach in calculating lower confidence limit and upper confidence limit as shown in picture below.

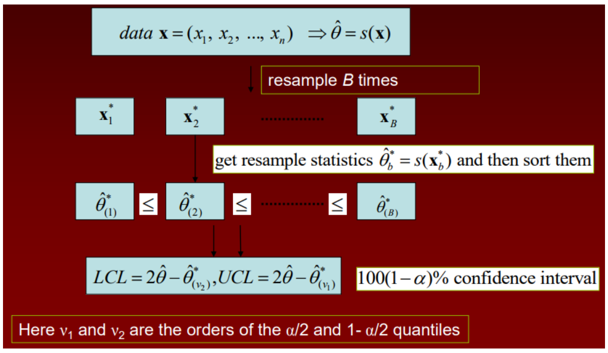

Figure 5: Step to Calculate CI Using Simple Method

Note that the length of interval is still same with the percentile or quantile method, but using this method, the center of interval is changed to be more symmetric.

Estimation of Confidence Interval for Standard Deviation

The second part is the calculation of confidence interval (CI) for standard deviation. Using the syntax below, then we can obtain the results for three methods (classical, quantile or percentile method, and simple method).

```
stan.dev <- sd(fci.calves)
n = length(fci.sort)
#set number of bootstrap samples
nsim=1000
stat = numeric(nsim) #create a vector in which to store the results
#Set up a loop to generate a series of bootstrap samples
for (i in 1:nsim){
  fciB = sample (fci.calves, n, replace=T)
  stat[i] = sd(fciB)}
#estimated standard error (classical to calculate CI)
#====================================================
mean(stat)
sd(stat)
##percentile method
#===================
quan <- quantile(stat,c(0.025,0.975))
boxplot (stat, main= "Boxplot␣of␣Feed␣Conversion␣Indexes␣of␣Calves␣-␣Botstrap")
hist(stat, main="Histogram␣-␣Precentile/Quantile␣Method")
abline(v=quan,col="red")
##simple method
#===============
a <- 2*stan.dev-quantile(stat,0.975)
b <- 2*stan.dev-quantile(stat,0.025)
hist(stat, main="Histogram␣-␣Simple␣Method")
abline(v=a,col="red")
abline(v=b,col="red")
```

The figure below shows the boxplot of the prediction of the standard deviation.

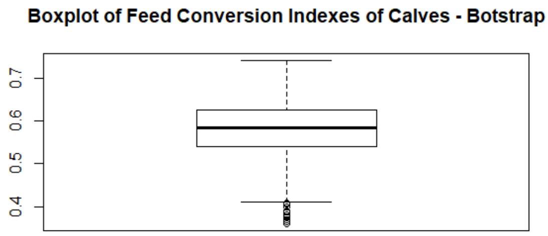

Figure 6: Boxplot of Bootstrap Simulation of Standard Deviation

Using classical method, it can be obtained the confidence interval (CI) which is 0.58 ± 0.12. For the second method, using quantile and percentile method, the result is 0.56 ± 0.13. And then histogram can be seen below, where the red line is the interval of confidence interval.

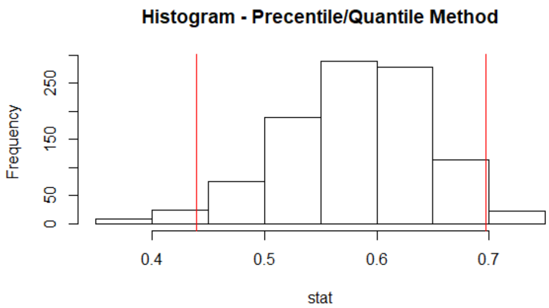

Figure 7: Histogram of Bootstrap Simulation - Quantile Method

Now, for the last method, using simple method, the result is 0.62 ± 0.13. And the following histogram shows where the confidence interval lies.

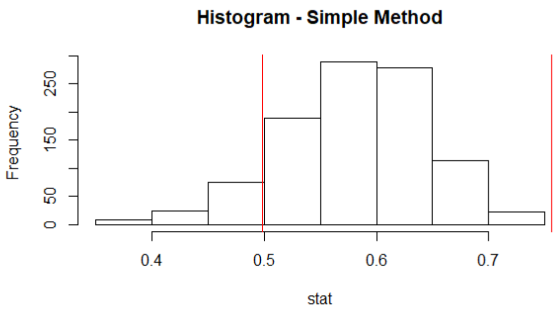

Figure 8: Histogram of Bootstrap Simulation - Simple Method

The same like the bootstraping method for the trimmed mean, for the standard deviation, both percentile method and simple method, they have the same range of interval, but different center, using the simple method, it makes it to be more symmetric.

Estimation of Confidence Interval for Coefficient of Variation

The last confidence interval will be calculated from statistics coefficient of variation. In probability theory and statistics, the coefficient of variation (CV), also known as relative standard deviation (RSD), is a standardized measure of dispersion of a probability distribution or frequency distribution. It is often expressed as a percentage, and is defined as the ratio of the standard deviation to the mean (or its absolute value).

Using the R syntax below, the results of confidence interval can be obtained directly, by the classical method, the interval of confidence interval is 0.16 ± 0.034.

```
coef.var <- sd(fci.calves)/mean(fci.calves)
n = length(fci.sort)
#set number of bootstrap samples
nsim=1000
stat = numeric(nsim) #create a vector in which to store the results
#Set up a loop to generate a series of bootstrap samples
for (i in 1:nsim){
  fciB = sample (fci.calves, n, replace=T)
  stat[i] = sd(fciB)/mean(fciB)}
#estimated standard error (classical to calculate CI)
#=====================================================
mean(stat)
sd(stat)
##percentile method
#==================
quan <- quantile(stat,c(0.025,0.975))
boxplot (stat, main= "Boxplot␣of␣Feed␣Conversion␣Indexes␣of␣Calves␣-␣Botstrap")
hist(stat, main="Histogram␣-␣Precentile/Quantile␣Method")
abline(v=quan,col="red")
##simple method
#==============
a <- 2*coef.var-quantile(stat,0.975)
b <- 2*coef.var-quantile(stat,0.025)
hist(stat, main="Histogram␣-␣Simple␣Method")
abline(v=a,col="red")
abline(v=b,col="red")
```

Below is the plot of boxplot that shows the results of the data obtained from bootstrap.

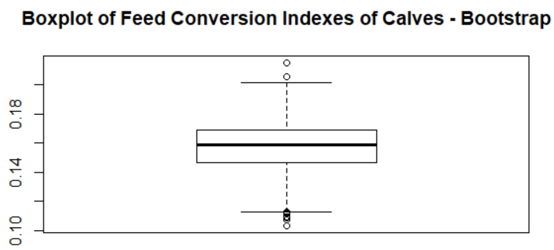

Figure 9: Boxplot of Bootstrap Simulation of Coefficient of Variation

The confidence interval obtained by percentile method is around 0.155 ± 0.033. And the following histogram illustrates the confidence interval of the data obtained from bootstrap.

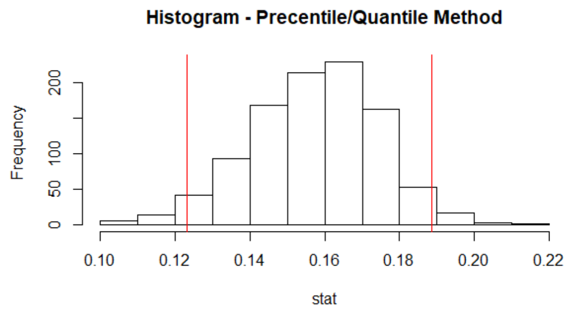

Figure 10: Histogram of Bootstrap Simulation - Quantile Method

And finally, using the simple method, the results of confidence interval is 0.168 ± 0.032. And to shows where the confidence interval lies, here is the histogram.

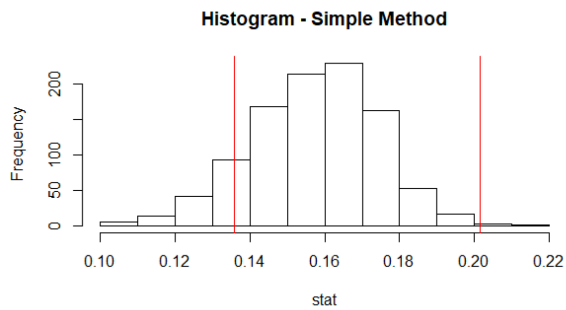

Figure 11: Histogram of Bootstrap Simulation - Simple Method

The same case with the previous two calculation of confidence interval, the best method to find the confidence interval from the bootstraped data is using simple method, because it has correction of the distribution (that make it symmetric) and the interval or range is still the same as in percentile or quantile method.

Problem Statement 2

We have a samples of 6 cars, of the same model and year, with the following prices, in a second-hand market, and total km:

Km	Prices
37388	14636
44758	14122
45833	14016
30862	15590
31705	15568
34010	14718

Table 1: Price of Cars

Calculate 95 % bootstrap CI (two methods) of the second hand price for a car (same model and year) with 50000 km.

Problem 2- Analysis

In order to find the prediction of the price of the cars, there are two models that will perform. The first model is exponential decay and the second one is second order polynomial.

First Model- Exponential Decay

From the problem above, the exponential decay model seems to be fit to the data. The exponential decay equation follows

$$y_i = a + e^{bx_i} + \epsilon_i$$

The figure 12 shows the fitting model to the data which is good enough to fit the data.

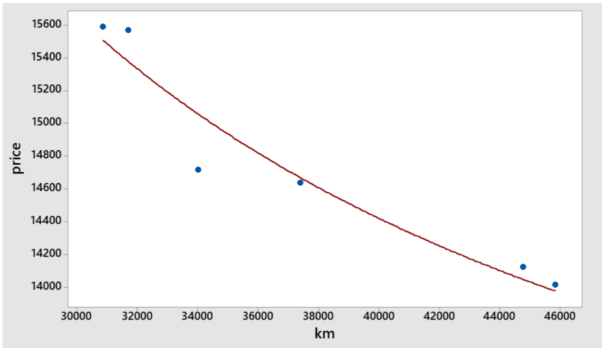

Figure 12: Fitting of the First Model

Now, the confidence interval for parameter a wil be claculated using three methods (classical method, percentile method, simple method, and bootstrap-t method).

Below is the R code to calculate the confidence interval of the parameter a and the prediction of the price of car at 50000 km.

```
km <- c(37388, 44758, 45833, 30862, 31705, 34010)
price <- c(14636, 14122, 14016, 15590, 15568, 14718)
car.data <- data.frame(km, price)
plot(car.data)
lm.car <- lm(log(price) ~ km)
summary(lm.car)
price.car <- exp(summary(lm.car)$coefficients[1]+summary(lm.car)$coefficients[2]*50000)
xx<-seq(30000,50000,by=10)
log.pred<-summary(lm.car)$coefficients[1]+summary(lm.car)$coefficients[2]*xx
plot(xx,log.pred)
plot(xx,exp(log.pred), main="Prediction of all the price of the cars with 30000-50000 km",
    xlab="km",ylab="price")

#####################################################################
n<-length(km);
data<- cbind(km, price);
nb<-1000;
z<-seq(1,n);
predb<-numeric(nb);
exp.predb <- numeric(nb);
t_predb <- numeric(nb);
exp.tpred <- numeric(nb);
for(i in 1:nb){
```

```
  zb <- sample(z,n,replace=T)
  lm.carb <- lm(log(data[zb,2])~ data[zb,1])
  predb[i]<- summary(lm.carb)$coefficients[1]+summary(lm.car)$coefficients[2]*
      50000
  exp.predb[i] <- exp(predb[i])
  sdpredb <- sd(predb)
  t_predb[i]<-(predb[i]-log.pred)/sdpredb
  exp.tpred[i] <- exp(t_predb[i])
}
#classical method-price cars
mean(exp.predb)
sd(exp.predb)
#quantile method-cars
quantile(exp.predb,c(0.025,0.975))
#simple
lower.limit.simple <- 2*price.car-quantile(exp.predb,0.975)
upper.limit.simple <- 2*price.car-quantile(exp.predb,0.025)
```

Figure 13. below shows the plot of the prediction of the data for cars with 30000 km until 50000 km.

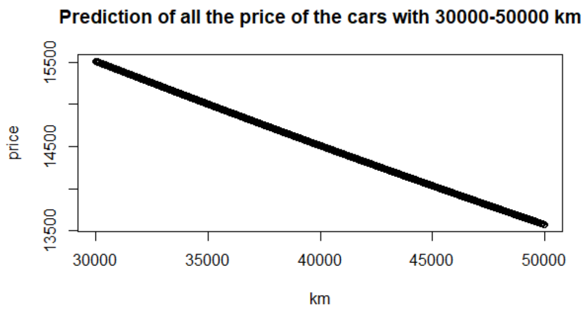

Figure 13: Prediction of the Price of the Cars Using the First Model

From the code above, the results can be obtained of some method, including quantile method and simple method.

Now, two methods used to calculate confidence interval will be evaluated and compared.

With the classical method, it can obtained the result is 13724.49 ± 2222.1, using quantile method, it can be seen that the interval is 14041.27 ± 1928.35, whereas using the simple method is 13108.8 ± 1928.35.

From the histogram below, it can be seen the interval for the two methods, where the red line is the confidence interval obtained by quantile method, the blue color is obtained with simple method.

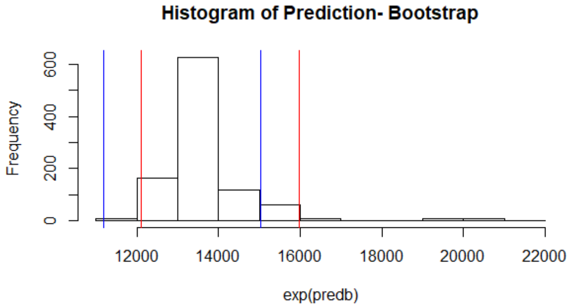

Figure 14: Histogram of the Prediction of Price using First Model

Second Model- Second Order Polynomial

The second model that may fit to the data is second order polynomial, with the equation

$$y_i = a + bx_i^2$$

where y is the price and x is the distance of the car (km).

The figure 15 below shows the prediction of the price of the car with the distance 30000-50000 km.

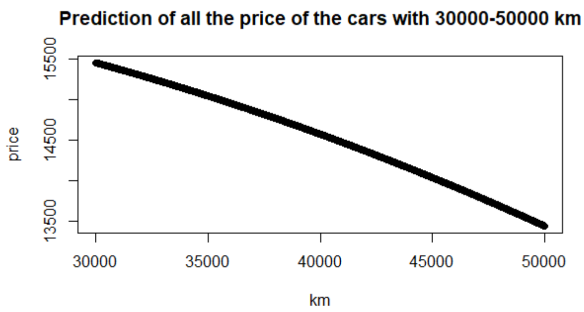

Figure 15: Prediction of the Car Price Using Second Model

Now, using the following code, the interval of confidence interval can be calculated.

```
km <- c(37388, 44758, 45833, 30862, 31705, 34010)
price <- c(14636, 14122, 14016, 15590, 15568, 14718)
car.data <- data.frame(km, price)
plot(car.data)
km2 <- km^2
pol.car <- lm(price ~ km2)
summary(pol.car)
price.car <- summary(pol.car)$coefficients[1]+summary(pol.car)$coefficients[2]*
    50000^2
xx<-seq(30000,50000,by=10)
pred<-summary(pol.car)$coefficients[1]+summary(pol.car)$coefficients[2]*xx^2
plot(xx,pred, main="Prediction of all the price of the cars with 30000-50000 km
    ",
      xlab="km",ylab="price")
############################################################################
n<-length(km);
data<- cbind(km, price);
```

```
nb<-1000;
z<-seq(1,n);
predb<-numeric(nb);
for(i in 1:nb){
  zb <- sample(z,n,replace=T)
  zb2 <- data[zb,1]^2
  pol.carb <- lm(data[zb,2] ~ zb2)
  predb[i]<- summary(pol.carb)$coefficients[1]+summary(pol.carb)$coefficients
      [2]*50000^2}
#classical method-price cars
mean(predb)
sd(predb)
#quantile method-cars
quantile(predb,c(0.025,0.975))
#simple
lower.limit.simple <- 2*price.car-quantile(predb,0.975)
upper.limit.simple <- 2*price.car-quantile(predb,0.025)
```

From the syntax above, the results can be analyzed, using classical method, it is obtained that the confidence interval is around 13299.3 ± 1331.5, using quantile method the result is around 12730.6 ± 1014.2, and using simple method, the result of confidence interval is around 14140.4 ± 1014.2. And figure below is the figure of histogram of the prediction of the price.

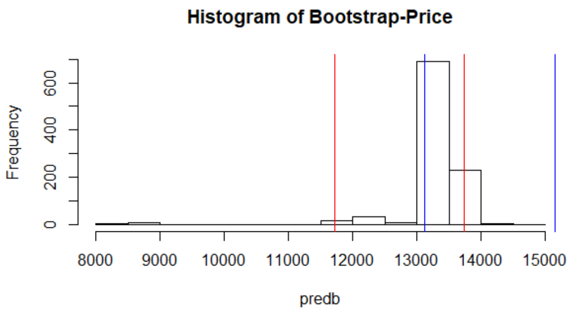

Figure 16: Histogram of Price Prediction Using Second Model

Comparison of Two Models

Now, two fitted model above can be compared. And the following table is the results from both two models

Methods	Exponential Model	Polynomial Model
Classical Methods	13724.49 ± 2222.1	13299.3 ± 1331.5
Quantile Methods	14041.27 ± 1928.35	12730.6 ± 1014.2
Simple Methods	13108.8 ± 1928.35	14140.4 ± 1014.2

Table 2: Comparison of the results of confidence interval from two models

Considering both the results presented in the table above and also histogram in figure 14 and 16, it shows that the first model is better than than the second model, since it fits to the data more.

Conclusion

From the problem two that has been analyzed, it can be concluded that: 1. The simple method of finding the confidence interval of the prediction (using bootstrap method) is better than classical method or quantile method, because it corrects the distribution (became symmetric), and the interval obtained is consistent with the quantile method. 2. The first model (exponential) is fitter to the data then the polynomial model.