Lab5

Andreas Stasinakis & Mim Kemal Tekin

February 18, 2019

Contents

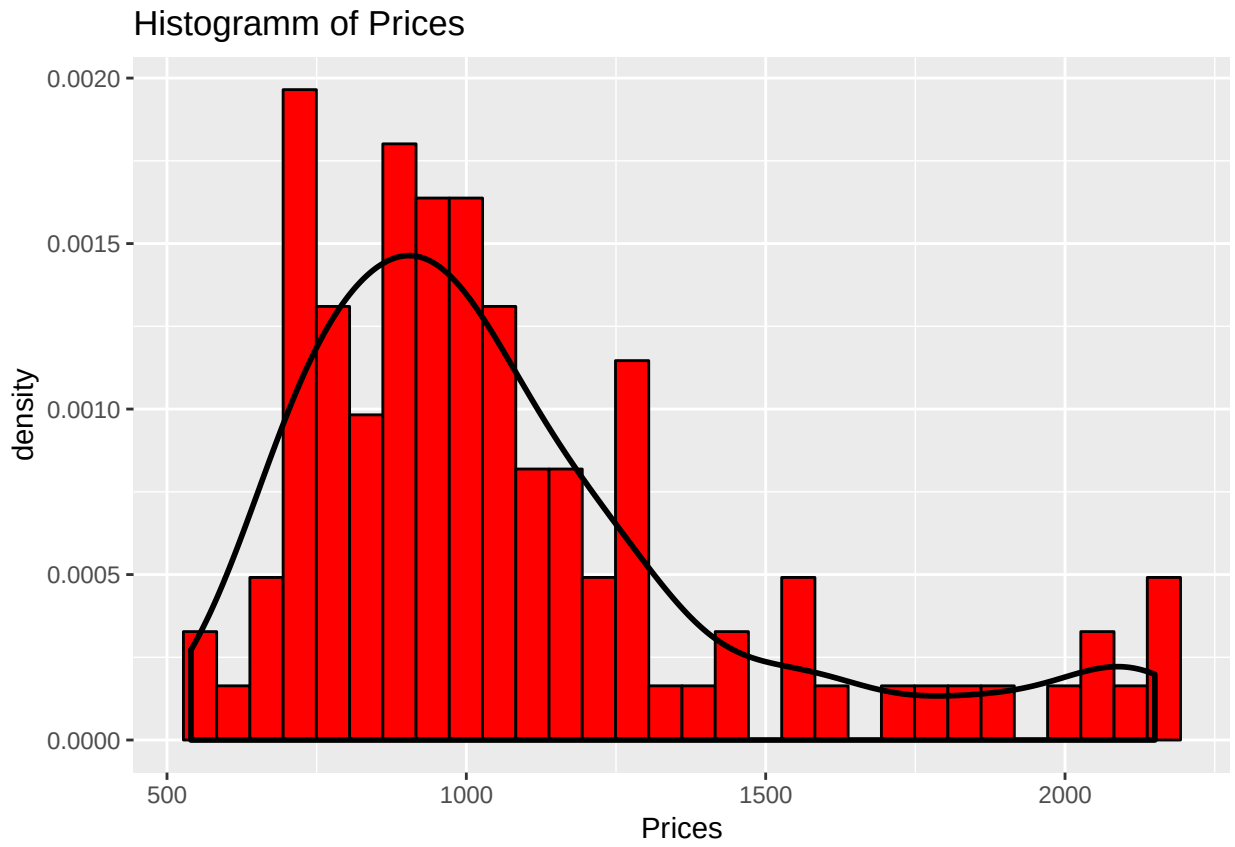
Question 2: Bootstrap, jackknife and confidence intervals

The data you are going to continue analyzing is the database of home prices in Albuquerque, 1993. The variables present are Price; SqFt: the area of a house; FEATS: number of features such as dishwasher, refrigerator and so on; Taxes: annual taxes paid for the house. Explore the file prices1.xls.

2.1 Histogramm of Price, discussion about the distribution of the data

Import Excel data, plot the histogram of Price. Does it remind any conventional distribution? Compute the mean price

```
library(ggplot2)
library(MASS)
set.seed(12345)
#import excel data
data = readxl::read_excel("../prices1.xls")

#Histogramm of Prices
ggplot(data) +
  geom_histogram(mapping = aes(data$Price,y = ..density..), color = "black", fill = "red", bins = 30)+
  geom_density(mapping = aes(data$Price), color = "black", size =1)+
  labs(title = "Histogramm of Prices", x = "Prices")
```




```
mean(data$Price)
```

```
## [1] 1080.473
```

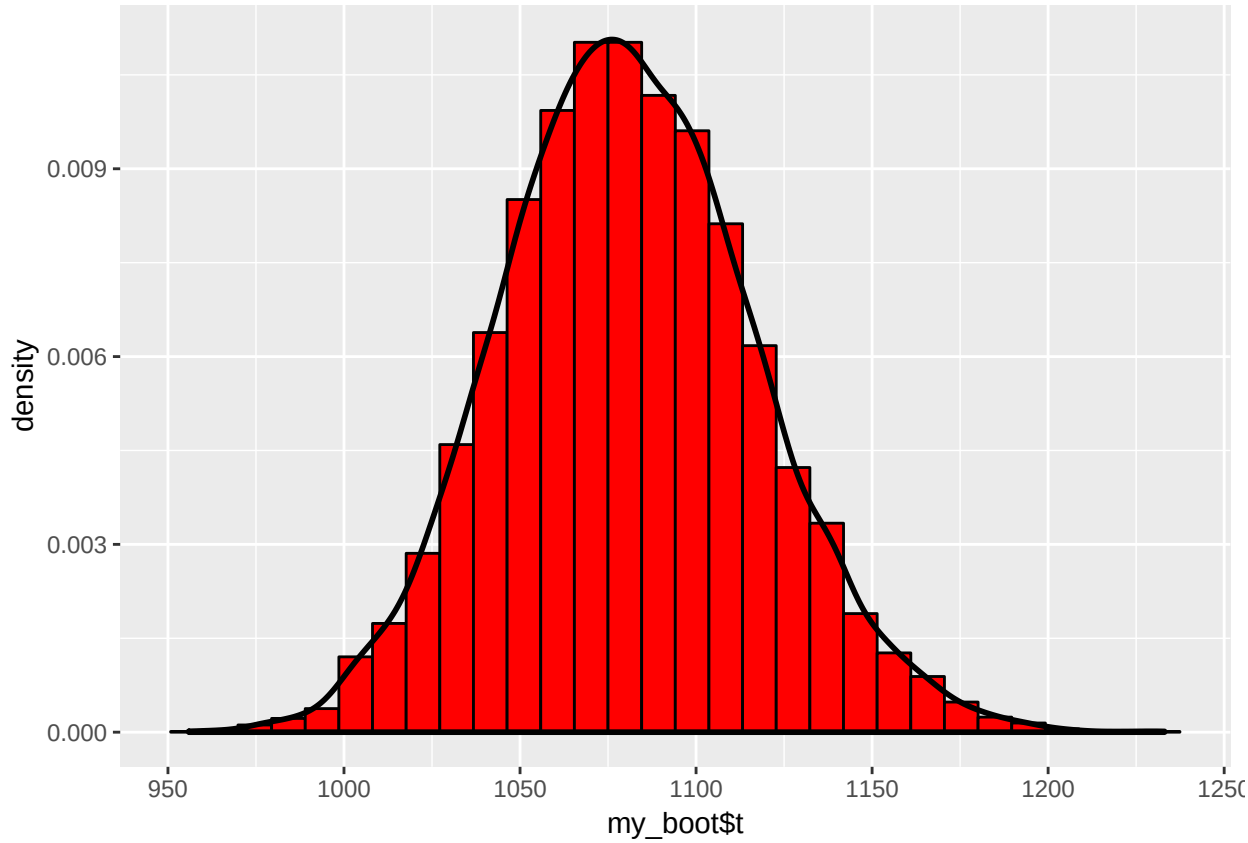
We can comment on the distribution of the value Y = Price from the data. First of all the distribution is right -tailed so right skewed. It seems like the variable Y follows Gamma or maybe a Log normal distribution. Of course we can not be sure 100% only by looking at the histogramm so we can make same Hypothesis testings or plot the probability plots and have a better picture.

2.2 Bootstrap bias-correction,variance, CI and first-order.

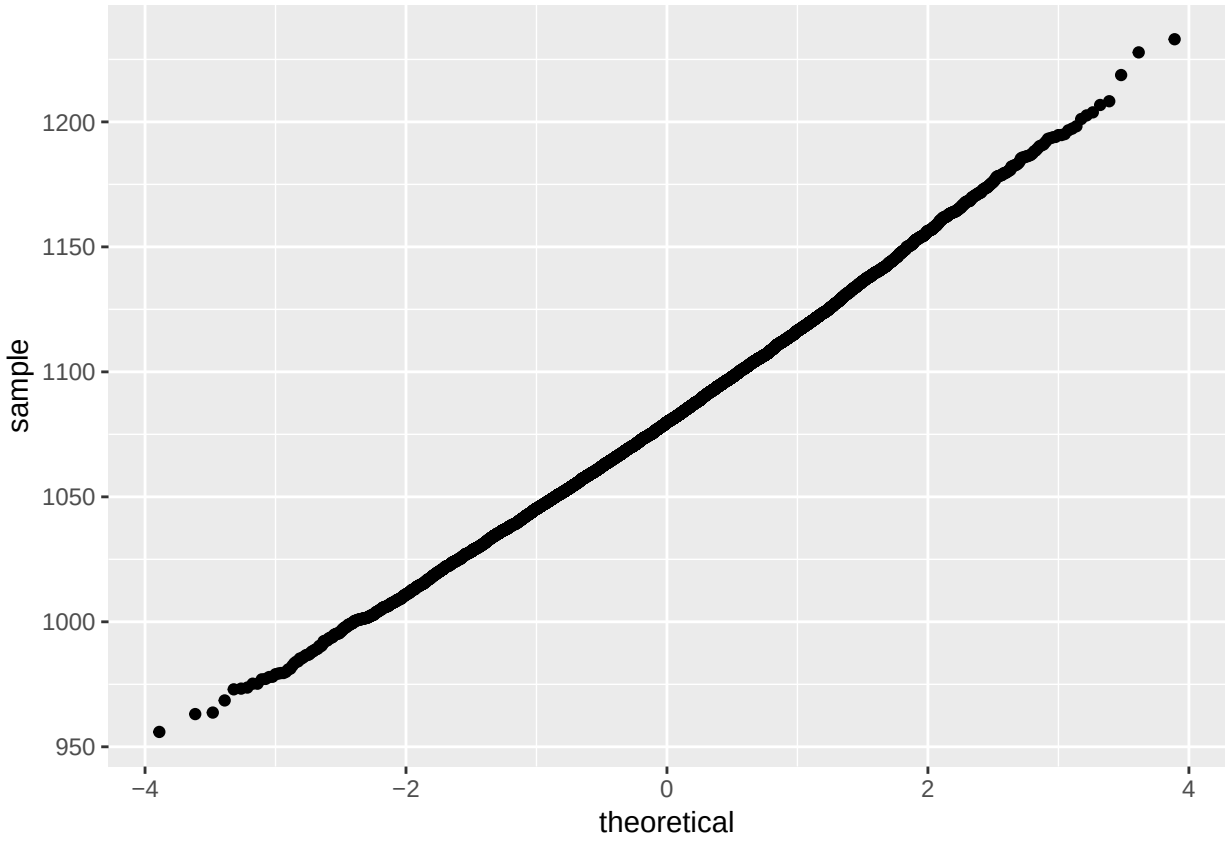
Estimate the distribution of the mean price of the house using bootstrap. Determine the bootstrap bias-correction and the variance of the mean price. Compute a 95% confidence interval for the mean price using bootstrap percentile, bootstrap BCa, and first-order normal approximation (Hint: use boot(),boot.ci(),plot.boot(),print.bootci())

```
library(boot)

# we need to create the statistic
# Statistic should take two inputs
# Data and index
my_mean = function(dt,ind){

  # returns the mean of each data set generated by bootstrap
  return(mean(dt[ind]))

}
```

```r
# use boot function to generate 1000 samples
my_boot = boot(data = data$Price,statistic = my_mean,R = 10000)

ggplot() +
  geom_histogram(mapping = aes(my_boot$t, ..density..),
                 bins = 30, color = "black", fill = "red")+
  geom_density(mapping = aes(my_boot$t, ..density..), color = "black", size =1)
```




```r
#Plot the quatile -quantile of Standar normals

ggplot(as.data.frame(my_boot$t), aes(sample=my_boot$t))+stat_qq()
```




```
# bias correction estimator
bias_cor = 2*mean(data$Price) - mean(my_boot$t)

#variance of estimator

var_est = sum((my_boot$t-mean(data$Price))^2)/(nrow(my_boot$t)-1)

#95% CI
# first order normal approximation, percentile interval,
# adjusted bootstrap percentile (BCa) interval
all_int = boot.ci(my_boot,type = c("norm","perc", "bca"))
all_int
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = my_boot, type = c("norm", "perc", "bca"))
##
## Intervals :
## Level      Normal             Percentile            BCa
## 95%   (1010, 1151 )   (1012, 1154 )   (1016, 1160 )
## Calculations and Intervals on Original Scale
```

In this task we want to estimate the distribution of the mean price of the houses using bootstrap(boot function). The boot function takes as an input the data, a statistic and the number of samples we want. The statistic is the function mean in this case. This function need two arquements as an input. Boot firstly

generates all the samples and after that uses that function for each sample. That is the reason why we have also the variable "ind"" in the function. We can see the histogramm of the estimated distribution and the quantiles for the normal distribution. In general we know from the Central limit theorem that in many cases, when we sum many independent random variables, this sum will follow a Normal distribution *EVEN* if the original variables are not distributed normally. So in this case, despite the fact that the random variable Y is *NOT* normaly distributed, the mean of those values is. From those two plots we can also confirm the assumption that the distribution of the mean price of the houses is Normal. The histogramm it is clear a Normal distribution with mean = 1080.6950927, really close to the mean of the Y distribution. From the quantile-quantile plot, if the data are normally distributed, most of the points should be on the line which bisect the two axis. This is also happening for this data, and it is one more proof that the mean of the Prices follows normal distribution.

We also calculate the bias correction estimator and the variance of our estimator.

The bias correction estimator is given by the formula

$$T = 2T(D) - \frac{1}{B}\sum_{1}^{B} T_i^*$$

,

where T() is the statistic function(in this case the mean). In this case the value of bias correction estimator is 1080.2503618.

??he variance of our estimator(the mean of the variable Price) using the formula :

$$\frac{1}{B-1}\sum_{i=1}^{B}\left(T(D_i^*) - \overline{T(D^*)}\right)^2$$

, where B is the number of bootstrap samples, $T(D_i^*)$ the statistic(mean) for each sample and $\overline{T(D^*)}$ is the mean of all the values after using the statistic for each sample. In this case the variance is equal to 1289.9893691. We can also calculate is from the boot function. It is the standard error of all samples with value same as using the formula.

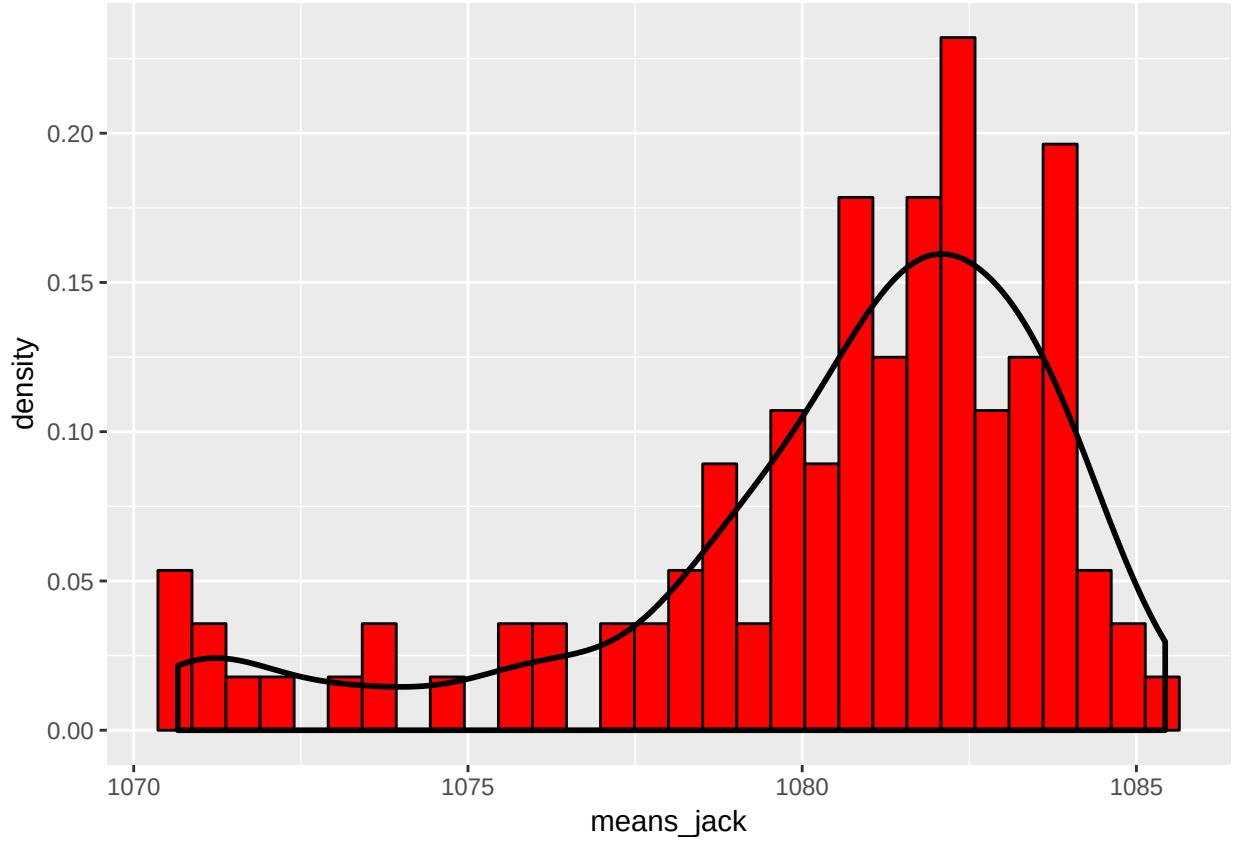
Finally, we compute and print 3 95% CI for the mean price using bootstrap percentile, bootstrap BCa and first order normal approximation.

2.3

Estimate the variance of the mean price using the jackknife and compare it with the bootstrap estimate

```
# function for generating jackknife samples
# input is the data we want to generate from
Jackknife = function(data){

  n = length(data)

  #matrix to store all samples
  samples = matrix(0,ncol = n-1, n)

  for (i in 1:n) {
    samples[i,] = data[-i]

  }

  return(samples)
}
```

```r
# generate the sample
jack_sample = Jackknife(data$Price)

# function using the statistic for each sample
# input: all the samples generated and a statistic function
Stat_jack = function(smp, statistic){

  # using apply function for calculate the statistic of each row(sample)

  results = apply(smp, 1,FUN = statistic)
  return(results)
}

# use the function to the sample and we have the distribution of the mean

means_jack = Stat_jack(jack_sample,mean)


# plot the distribution of the samples
ggplot() +
  geom_histogram(mapping = aes(means_jack, ..density..),
                 bins = 30, color = "black", fill = "red")+
  geom_density(mapping = aes(means_jack, ..density..), color = "black", size =1)
```



```r
# function compute the variance estimator for Jackknife sample
n = length(means_jack)

a = (n-1)*means_jack
b = n*mean(data$Price)
T_i = b - a

J_t = mean(T_i)

#Final variance estimator
Jack_var = (sum((T_i-J_t)^2))/(n*(n-1))
```

In this task we have to estimate the variance of the mean price using the Jackknife algorithm. A short description of the algorithm is presented. We generate n samples as the number of observations we have(110 here). The $i-th$ sample is the hole data except for the i observation. For example the 5th sample is the data except for the X_5. After we have the samples, we use the statistic(in this case the mean) and with this way we estimate the distribution of the mean price. Finally, in order to calculate the variance estimator we use the formula below:

$$\hat{Var[T()]} = \frac{1}{n(n-1)} \sum_{i=1}^{n} \Big(T_i^* - J(T)\Big)^2$$

, where $T_i^* = nT(D) - (n-1)T(D_i^*)$, $J(T) = \frac{1}{n}\sum_{i=1}^{n} T_i^*$ and $n = 110$.

The value we obtain using the Jackknife algorithm is 1320.9110441 , while for the bootstrap we have 1289.9893691. The two algorithms do not have many differences. Bootstrap generates samples with replacement, in contrast to Jackknife which leaves one observation out in each iteration(much less computational heavy). Therefore, Jackknife we always have almost the same sample and the number of this samples is the same as the observations. Therefore the variance estimation is always the same. On the contrary, we can use bootstrap algorithm in order to generate as many samples as we want, and every time we may have different samples. Moreover, we can observe that the variance estimation using Jackkife is higher than the bootstrap method. This is happening because Jackknife is a more convervative method than Bootstrap as a result it is overestimates the variance.

2.4 Compare the CI from bootstrap

Compare the confidence intervals obtained with respect to their length and the location of the estimated mean in these intervals.

We can see the results for each confidence interval in the table below:

```r
Normal = c(1010,1151,141,1080.5)
percentile = c(1012,1154,142,1083)
BCa = c(1016,1160,144,1088)
int = rbind(Normal,percentile,BCa)

colnames(int) = c("From","To  ","Range"  ,"Mean")

knitr::kable(x = int, caption = "Compare different confidence intervals")
```

Table 1: Compare different confidence intervals

	From	To	Range	Mean
Normal	1010	1151	141	1080.5
percentile	1012	1154	142	1083.0

	From	To	Range	Mean
BCa	1016	1160	144	1088.0

In general it can be said that Normal and the percentile interval are first- order accurate while BCa interval are second-order accurate. In this case the intervals above are really similar. Normal starts from 1010, percentile from 1012 and BCa from 1016. All of them end between 1151-1160. Their ranges are almost the same. The smallest one is the Normal and the largest the BCa.