Question 1: Hypothesis testing

In 1970, the US Congress instituted a random selection process for the military draft. All 366 possible birth dates were placed in plastic capsules in a rotating drum and were selected one by one. The first date drawn from the drum received draft number one, the second date drawn received draft number two, etc. Then, eligible men were drafted in the order given by the draft number of their birth date. In a truly random lottery there should be no relationship between the date and the draft number. Your task is to investigate whether or not the draft numbers were randomly selected. The draft numbers (Y=Draft No) sorted by day of year (X=Day of year) are given in the file lottery.xls.

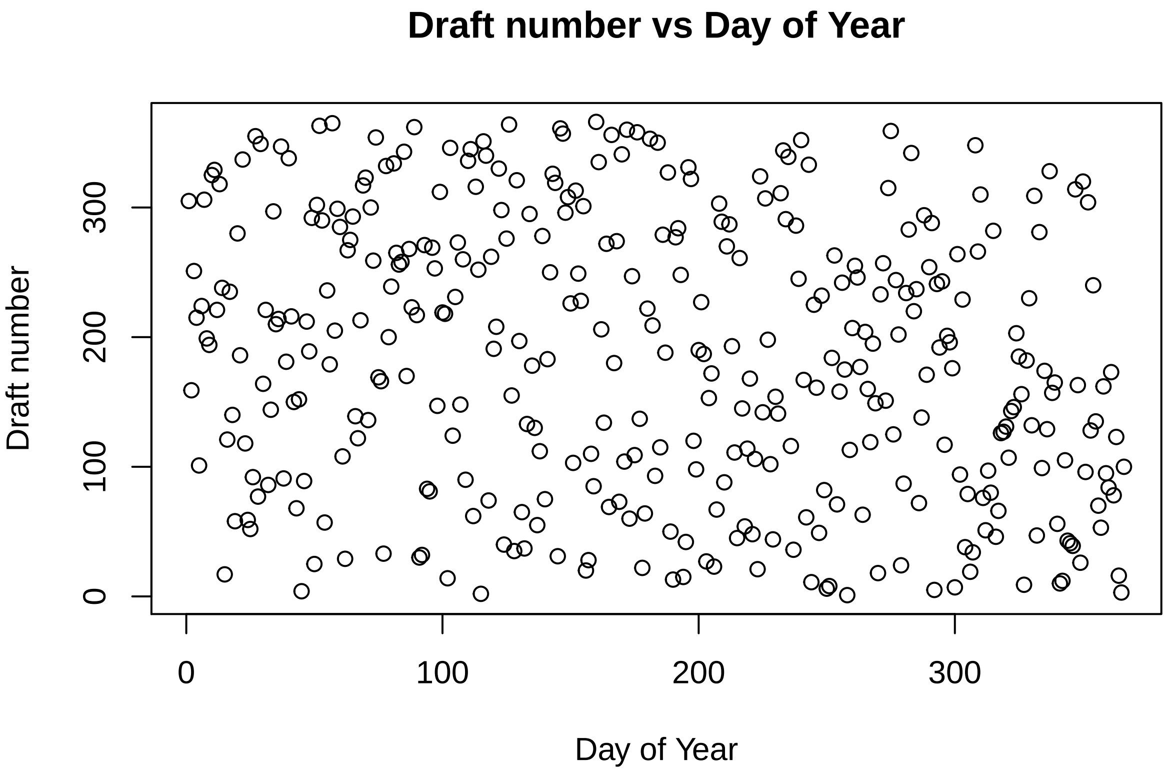
1. Make a scatterplot of Y versus X and conclude whether the lottery looks random.

```
RNGversion("3.5.1")
```

```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used
```

```
set.seed(12345)
library(readxl)

data <- read_excel("C:/Users/Vaso/Desktop/data computational/lottery.xls")
df <- data.frame("X"=data$Day_of_year,"Y"=data$Draft_No)
```

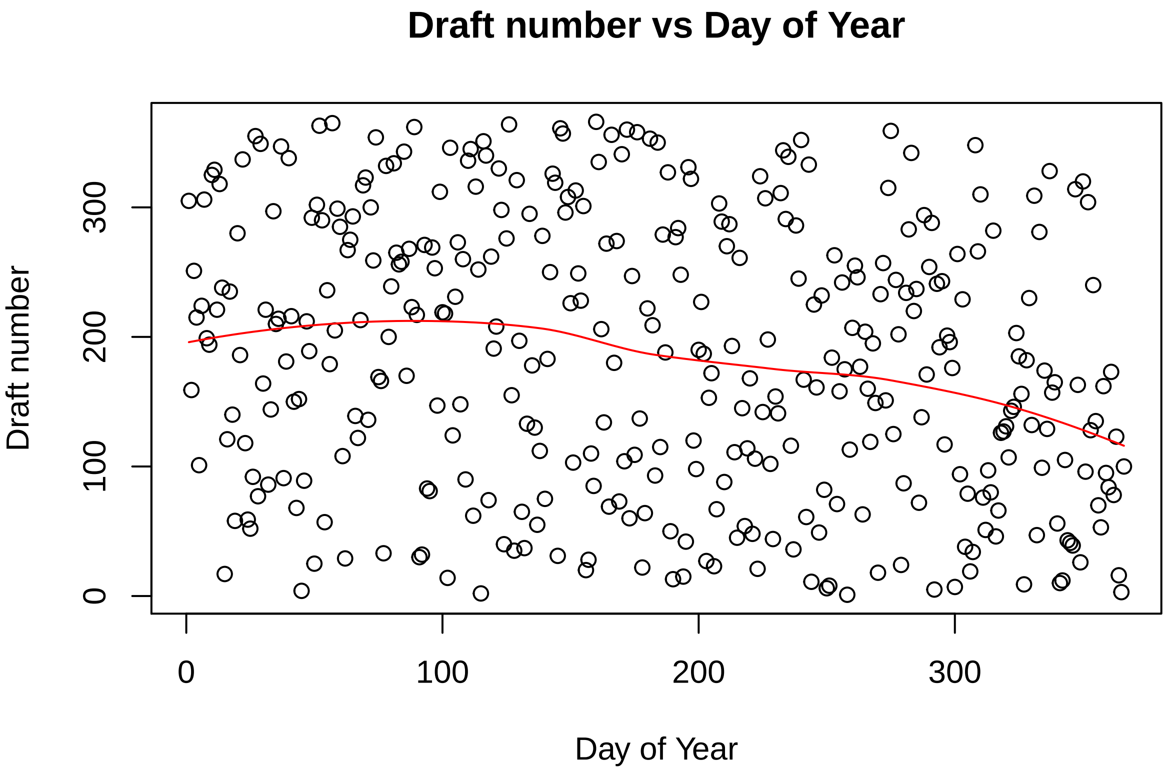
```
plot(df$X,df$Y,main="Draft number vs Day of Year",xlab="Day of Year"
     ,ylab="Draft number")
```



As we can see from the scatterplot, it seems that there isn't any specific pattern of the points so we can conclude that the lottery looks random and there's no relationship between the date and the draft number.

2. Compute an estimate $\hat{Y}$ of the expected response as a function of X by using a loess smoother (use loess()), put the curve $\hat{Y}$ versus X in the previous graph and state again whether the lottery looks random.

```
model <- loess(Y~X,data=df)
plot(df$X,df$Y,main="Draft number vs Day of Year",xlab="Day of Year"
     ,ylab="Draft number")
lines(df$X,model$fitted,col="red")
```



Plotting the **Expected** Draft Number,this time, against X we observe that the lottery doesn't look random as there is a downward trend. More precisely, it seems that the highest draft numbers are obtained for the first days of the year and as days go by, the expected draft numbers become smaller,steadily. So it seems as there is a reverse relation between X and $\hat{Y}$.

3. To check whether the lottery is random, it is reasonable to use test statistics T $= \frac{Y(\hat{X}_b)-Y(\hat{X}_a)}{X_b-X_a}$ **where** $X_b = argmax_X\hat{Y}(X)$**,** $X_a = argmin_X\hat{Y}(X)$**. If this value is significantly greater than zero, then there should be a trend in the data and the lottery is not random. Estimate the distribution of T by using a non-parametric bootstrap with B = 2000 and comment whether the lottery is random or not. What is the p-value of the test?**

We will use the test statistics $T = \frac{Y(\hat{X}_b)-Y(\hat{X}_a)}{X_b-X_a}$ where

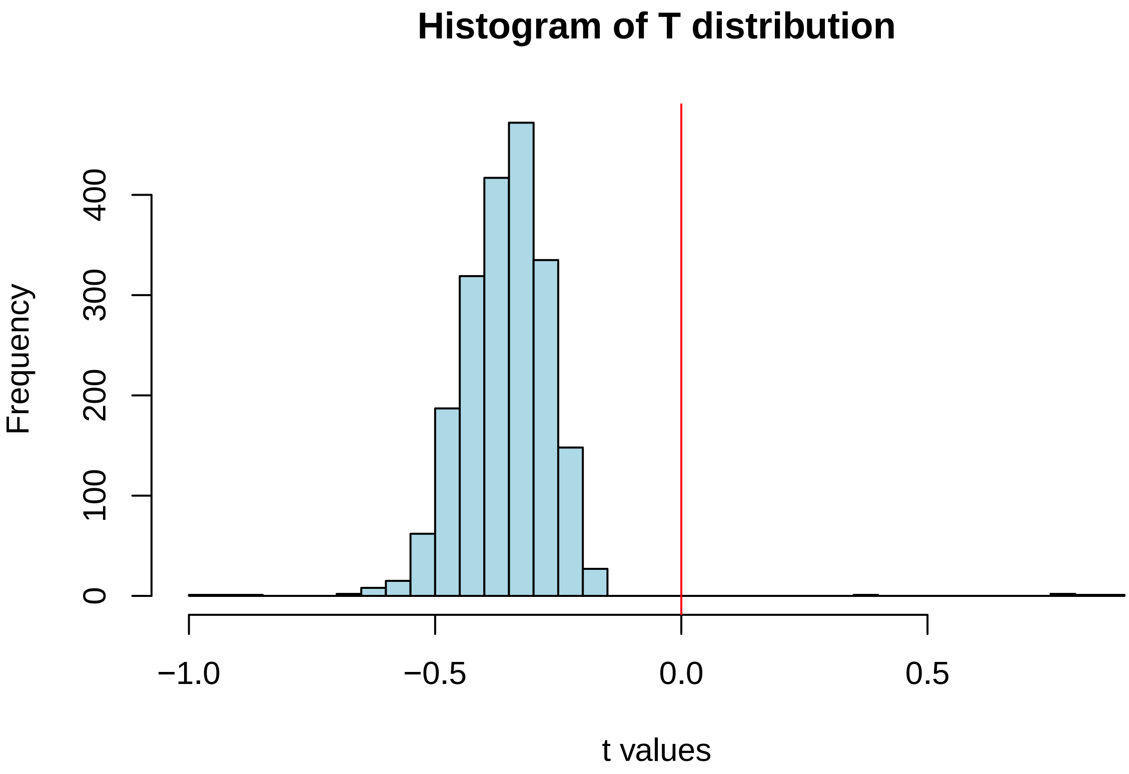
$X_b = argmax_X\hat{Y}(X)$, $X_a = argmin_X\hat{Y}(X)$ to check whether the lottery is random.

So the hypothesis test we want to conduct is:

H_0 : T=0 lottery is random vs. H_1 : T$\neq$ 0 : lottery is not random. (NOT SURE)

For this, we estimate the distribution of T using a non-parametric bootstrap with B=2000. That means that from the original data we generate 2000 different samples with replacement.

```r
library(boot)
set.seed(12345)

stat <- function(data, ind) {
  d2 <- as.data.frame(data[ind,]) # to select sample
  model2 <- loess(Y~X,data=d2)
  Yhat <- model2$fitted
  Xa <- d2$X[which.min(Yhat)]
  Xb <-  d2$X[which.max(Yhat)]
  Y_a <- min(Yhat)
  Y_b <- max(Yhat)
  T= (Y_b - Y_a)/(Xb-Xa)
  return(T)
}



res=boot(df,stat,R=2000)
hist(res$t,breaks = 40, main="Histogram of T distribution",xlab="t values",
     col="light blue")
abline(v=0,col="red")
```



Observing the histogram of T, we can see that almost all values of the T statistics is below 0. Thus, we wouldn't say that the T statistics is significantly greater than zero, so we reject the null hypothesis, the lottery is not random.

We now want to compute the p-value which is the probability of an observed or very unlikely(because of the tail) result under the assumption that the null hypothesis is true.

p-value=

$$\frac{\#|T(X_i)| > T(X)}{B}$$

In our case, p-value=0.0025 <0.05 which means that we reject the null hypothesis. The lottery is not random.

```
## [1] 0.0025
```

4. Implement a function depending on data and B that tests the hypothesis H0: Lottery is random versus H1: Lottery is non-random by using a permutation test with statistics T. The function is to return the p-value of this test. Test this function on our data with B = 2000.

We want to test the hypothesis:

H_0: Lottery is random vs. H_1: Lottery is non-random by using a permutation test and T statistics $T = \frac{Y(\hat{X_b})-Y(\hat{X_a})}{X_b-X_a}$,where

$X_b = argmax_X \hat{Y}(X)$, $X_a = argmin_X \hat{Y}(X)$.

The difference between bootstrap and permutation tests is that in bootstrap resamples are drawn with replacement whereas in permutation test resamples are drawn without replacement.

```r
set.seed(12345)
permutation <- function(data,B){
  stat=numeric(B)
  n=dim(data)[1]
  for (i in 1:B){
    data$Y <- sample(data$Y,n)
    model2 <- loess(Y~X,data=data)
    Yhat <- model2$fitted
    Xa <- data$X[which.min(Yhat)]
    Xb <-  data$X[which.max(Yhat)]
    Y_a <- min(Yhat)
    Y_b <- max(Yhat)
    stat[i]= (Y_b - Y_a)/(Xb-Xa)
  }
 stat0 <- (Y_b - Y_a)/(Xb-Xa)
 return(list("stat"=stat0 ,"p-value"=mean(stat>stat0)))
}
permutation(df,2000)
```

```
## $stat
## [1] -0.1341398
##
## $`p-value`
## [1] 0.669
```

Based on the p-value which is equal to 0.669>0.05 we fail to reject the null hypothesis, so based on this test Lottery is random.

5. Make a crude estimate of the power of the test constructed in Step 4:

(a) Generate (an obviously non-random) dataset with n = 366 observations by using same X as in the original data set and Y(x)= max(0, min(αx + β, 366)), where α = 0.1 and $\beta \sim$ N(183,sd = 10).

```
set.seed(12345)
newX <- df$X
newY <- vector(length=366)
b <- rnorm(366,183,10)

for (i in 1:366){
  newY[i] <- max(0, min(0.1*newX[i]+b[i],366))
}

newdata <- data.frame("X"= newX,"Y"= newY)
```

(b) Plug these data into the permutation test with B = 200 and note whether it was rejected.

```
permutation(newdata,200)
```

```
## $stat
## [1] 0.01251306
##
## $`p-value`
## [1] 0.415
```

Using the new dataset that we constructed we get 0.415>0.05 as p-value and thus we fail to reject the null hypothesis, so based on this test Lottery is random.

(c) Repeat Steps 5a-5b for α= 0.2, 0.3, . . . ,10.

```
X_rep <- df$X
b <- rnorm(366,183,10)
a<- seq(0.2,10,0.1)
```

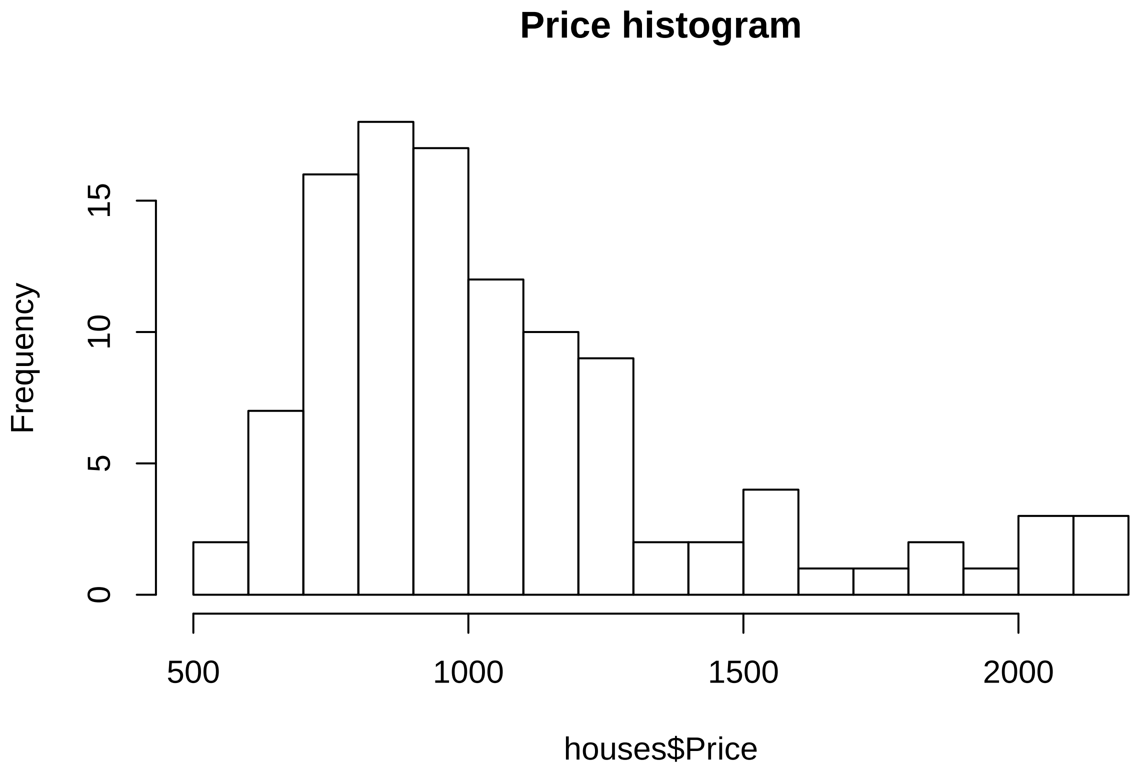
Question 2: Bootstrap, jackknife and confidence intervals

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.

1. Plot the histogram of Price. Does it remind any conventional distribution? Compute the mean price.

```
houses <- read_excel("C:/Users/Vaso/Desktop/data computational/prices1.xls")
```

```
hist(houses$Price,breaks=20,main="Price histogram")
```



```
mean(houses$Price)
```

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

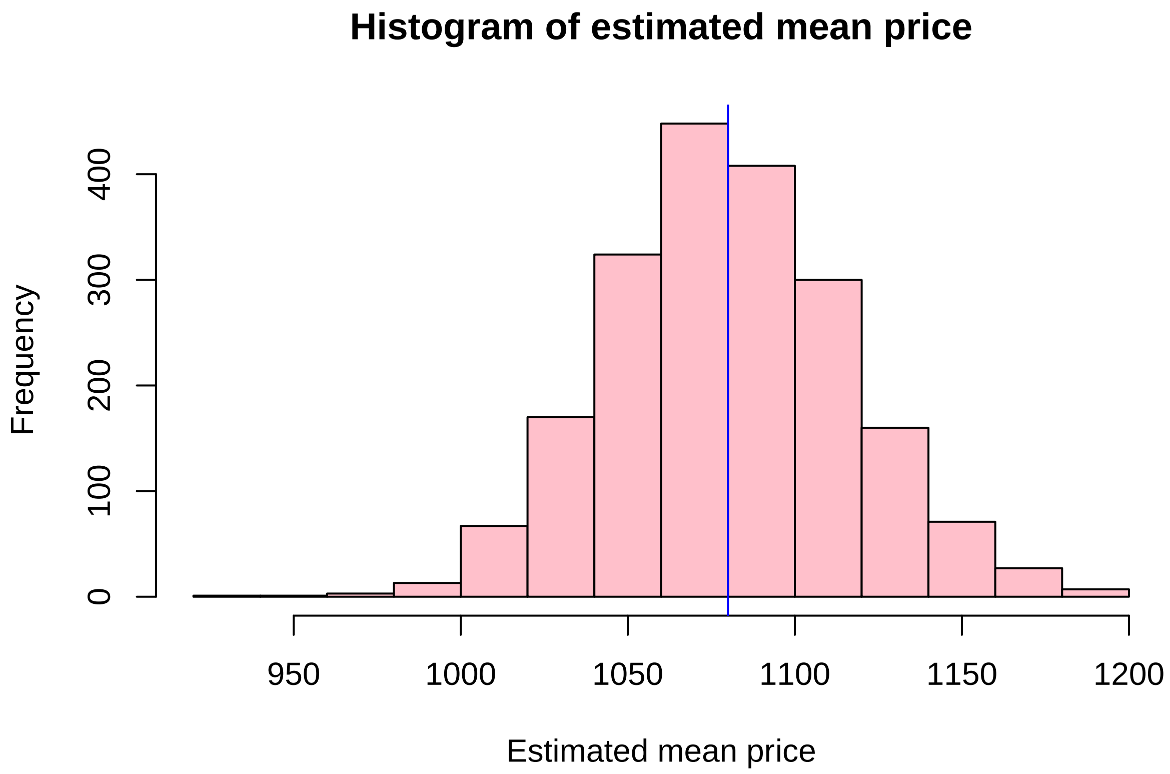
Looking at the histogram of Prices we wouldn't say that it reminds us any of the conventional distributions as the shape of the histogram is not clear.

The mean price is 1080.473.

2. 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()).

```
set.seed(12345)

stat <- function(data, ind) {
  d <- as.data.frame(data[ind,]) # to select sample
  stat0 <- mean(d$Price)
}

res1=boot(houses,stat,R=2000)

estim_mean <- mean(res1$t) #mean of the estimated mean prices
```

```
hist(res1$t,main = "Histogram of estimated mean price",col="pink"
     ,xlab="Estimated mean price")
abline(v=estim_mean,col="blue")
```



Based on the above histogram we could say that the mean price is normally distributed. The blue vertical line represents the mean of the estimated mean prices,which is 1079.983.

The bias correction estimator is $2T(D) - \frac{1}{B}\sum_{i=1}^{B} T_i^*$. Using R to calculate this we obtain:

```
T1 = 2*res1$t0 - mean(res1$t)
cat("The bias corrections is:",T1,"\n")
```

```
## The bias corrections is: 1080.963
```

?????? TOO CLOSE TO REAL VALUE

```
B <- 2000
variance  <- sum(res1$t - mean(res1$t))^2 / (B-1)
cat("The variance of the estimator is:",variance,"\n")
```

```
## The variance of the estimator is: 4.221145e-24
```

The variance of the estimator is very small which means the uncertainty is very low, and the estimator seems to be accurate.

Another measure of uncertainty is the confidence intervals. We will compute the different types of confidence intervals using the boot.ci() function. Let's see some theory first. (http://users.stat.umn.edu/~helwig/notes/bootci-Notes.pdf)

Bootstrap Confidence Interval via Percentiles:

For this type of CI the 100(1-α)-th percentiles of $T(Di^*)$ are used. So, for a 95% confidence interval, in our case that we have B=2000, the CI will be: $[T^*_{50}, T^*_{1950}]$. *Note:* All the $T(Di^*)$ must be sorted first.

Bootstrap BCa CI:

Those intervals use percentiles of bootstrap distribution, but they do not necessarily use the 100(1-α)-th percentiles.

First-order normal approximation Bootstrap CI:

Those intervals are computed using $\hat{\sigma_B}$ which is the sd of the bootstrap samples and $\bar{T}$ which is the mean. So, the 95% CI is: $\bar{T} \pm z_{\frac{\alpha}{2}}\hat{\sigma_B}$.

```r
print(boot.ci(res1))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = res1)
##
## Intervals :
## Level      Normal              Basic
## 95%   (1011, 1151 )   (1009, 1148 )
##
## Level     Percentile            BCa
## 95%   (1013, 1152 )   (1018, 1162 )
## Calculations and Intervals on Original Scale
```

Running the boot.ci() function we got those results:

Bootstrap Confidence Interval via Percentiles: (1013, 1152)

Bootstrap BCa CI: (1018, 1162)

First-order normal approximation Bootstrap CI: (1011, 1151)

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

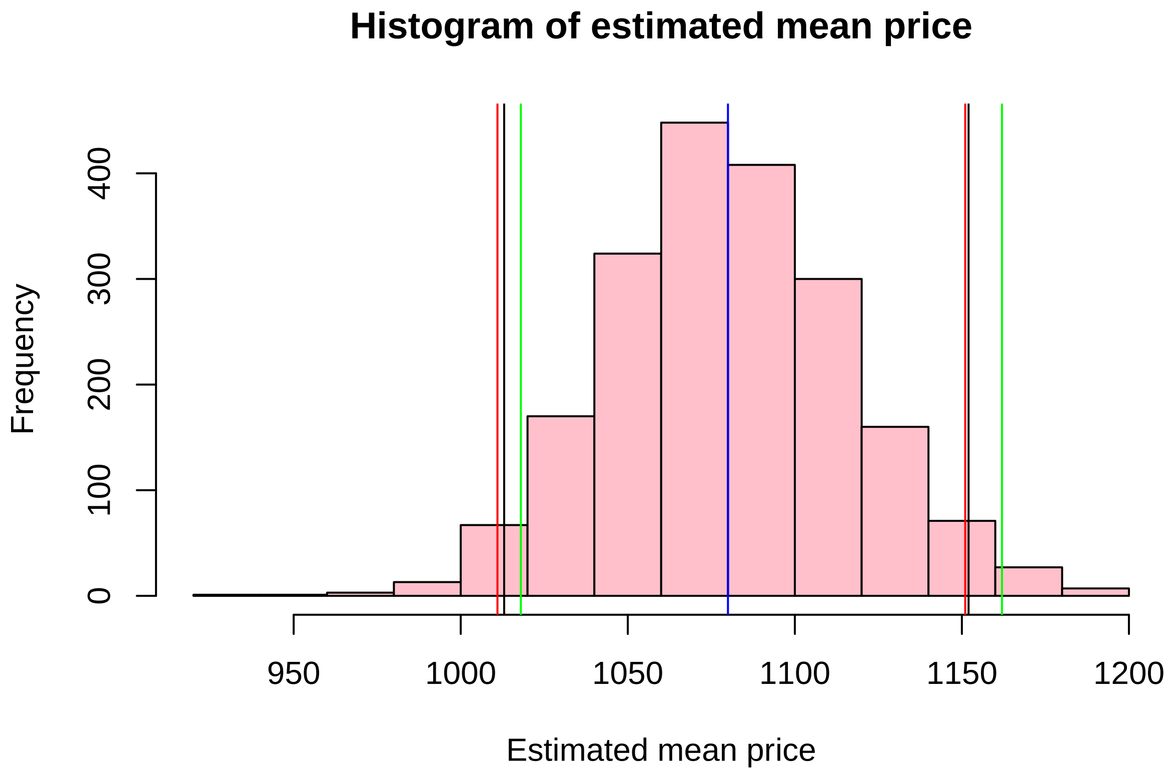
```r
#n<=B for jackknife
set.seed(12345)
n <- dim(houses)[1]
T <- rep(0,n)

for (i in 1:n){
  T[i] = n*mean(houses$Price) - (n-1)*mean(houses$Price[-i])
}

var_jack = sum(T-mean(T))^2 / (n*(n-1))
var_jack
```

```
## [1] 9.934446e-27
```

Using the jackknife algorithm the variance of the mean price is 9.934446e-27 and it's smaller than the variance obtained using the bootstrap estimate(4.221145e-24). Thus, the uncertainty of the estimator obtained using Jackknife is smaller compared to Bootstrap method.

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

```r
hist(res1$t,main = "Histogram of estimated mean price",col="pink"
     ,xlab="Estimated mean price")
abline(v=estim_mean,col="blue")
abline(v=c(1013, 1152)) # Percentiles CI
abline(v=c(1018, 1162 ),col="green") #BCa CI
abline(v=c(1011, 1151 ),col="red") #First-order normal CI
```



```r
#Length of the CIs:
len_per <- abs(1013-1152)
len_BCa <- abs(1018-1162)
len_norm <- abs(1011-1151)
```

Above, we have the histogram of estimated mean prices using bootstrap and the blue line represents the mean of all those estimated mean prices.

Bootstrap Confidence Interval via Percentiles: (1013, 1152) and it's represented by the **black** lines.

Bootstrap BCa CI: (1018, 1162) and it's represented by the **green** lines.

First-order normal approximation Bootstrap CI: (1011, 1151)and it's represented by the **red** lines.

First of all we observe that the estimated mean(1079.983) falls inside all of the three different confidence intervals. Considering the length of the CIs the wider one is the BCa(length=144) and the narrowest one is the percentiles CI(length=139). The First-order normal CI has length 140.

However, it seems that all the three confidence intervals have more or less the same length and no significant differences can be detected between them.