Computational Statistics

Lab 5

Emil K Svensson and Rasmus Holm

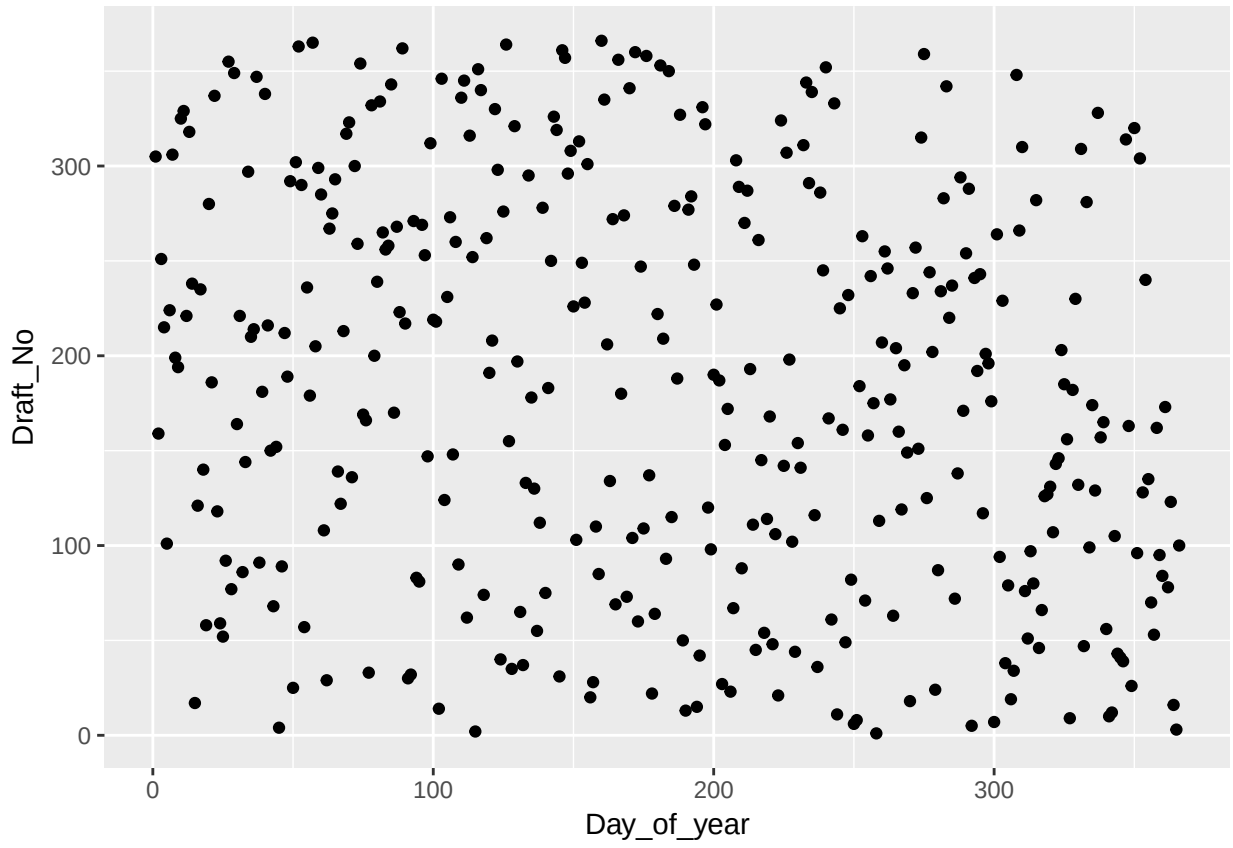
2017-03-09

Question 1

In this exercise we are given a data set of a random selection process for the military draft and we are supposed to use non-parametric bootstrap and permutation testing to test the null-hypothesis that the selection process was actually random.

1.1

```
library(ggplot2)

lottery <- read.csv2("../data/lottery.csv")

q11 <- ggplot(lottery, aes(x = Day_of_year, y = Draft_No)) + geom_point()
plot(q11)
```

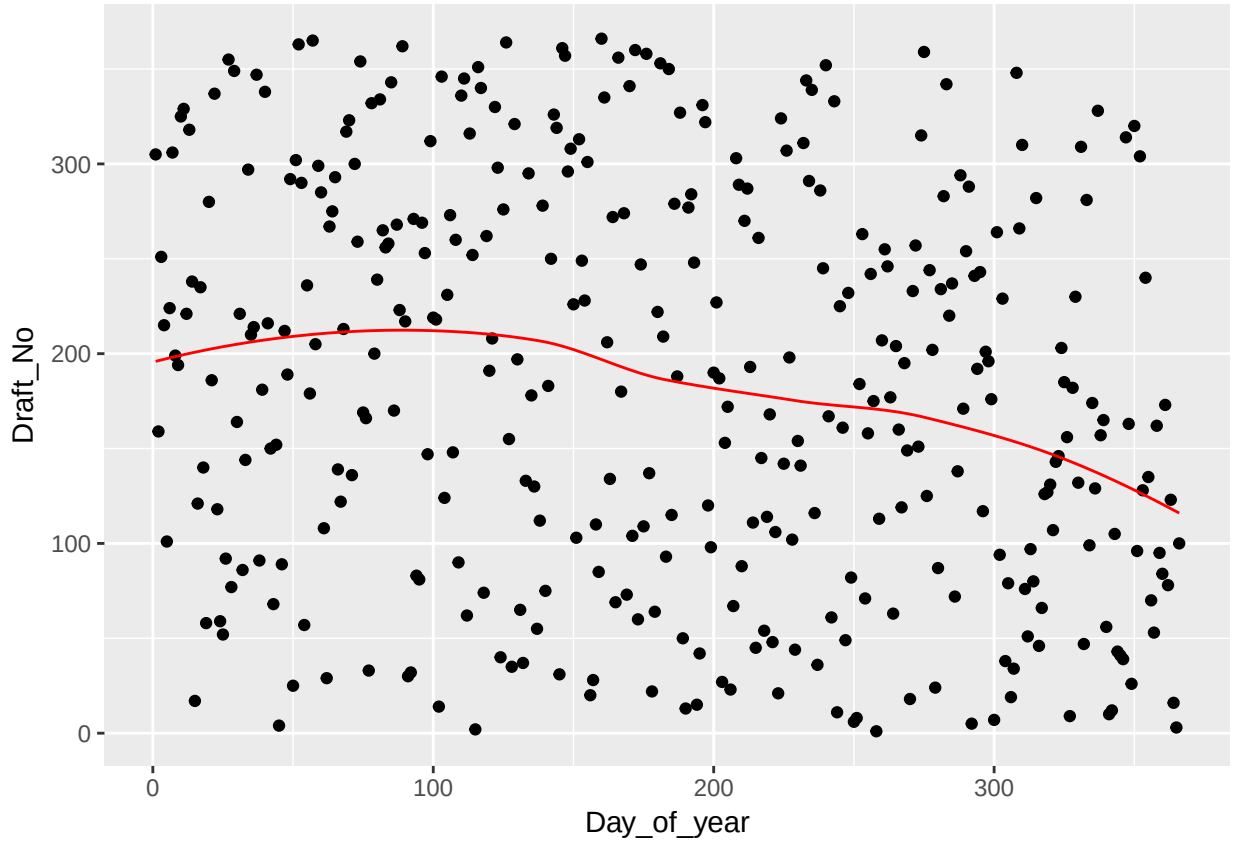


```
data <- data.frame(x=lottery$Day_of_year, y=lottery$Draft_No)
```

The data looks random although there might be some sort of skewness in the right side of the graph where there are lacking some observations in the top part and therefore men born within the last 3 months of the year had a higher probability of getting drafted.

1.2

To investigate the skewness further we used local regression which we would expect to be completely straight if the data is random. Below is a plot of the fit.



The fit (line) is not straight and have a decreasing trend which would support previous statement that men born on a day later in the year have a higher probability of being drafted.

1.3

To test that the lottery is random we first used non-parametric bootstrapping with the following test statistics

$$T = \frac{\hat{Y}(X_b) - \hat{Y}(X_a)}{X_b - X_a},$$

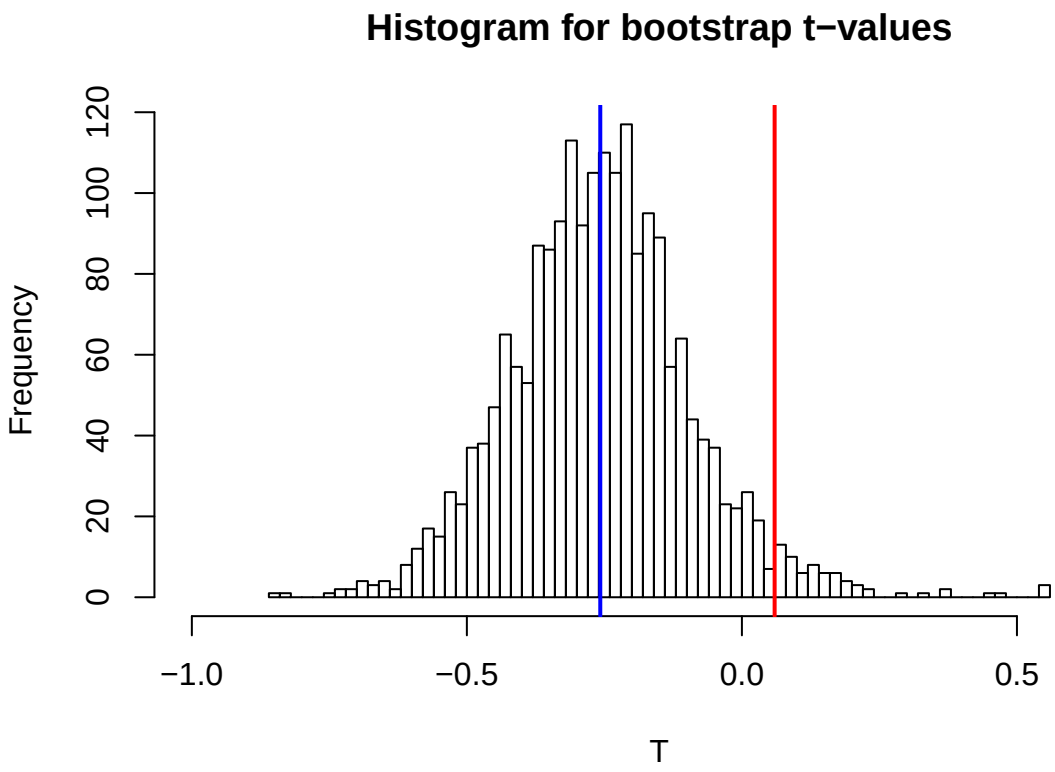
where

$$X_a = \underset{X}{\operatorname{argmin}}\, Y(X),$$
$$X_b = \underset{X}{\operatorname{argmax}}\, Y(X),$$

and $\hat{Y}(X)$ is the estimate from local regression. Observe that the test statistics tells us that if it is significantly larger than zero then the data is not random, that there is a trend in the data. We used 2000 bootstrap samples to estimate the distribution of T and the result is presented below.

```
teststat <- function(model) {
    function(data) {
        xa <- data$x[which.min(data$y)]
        xb <- data$x[which.max(data$y)]

        fit <- model(y ~ x, data)

        ya <- predict(fit, xa)
        yb <- predict(fit, xb)

        (yb - ya) / (xb - xa)
    }
}

teststat_boot <- function(data, idx, stat) {
    data <- data[idx,]
    stat(data)
}
```

```
library(boot)

B <- 2000

set.seed(123456)
npboot <- boot(data=data, statistic=teststat_boot, R=B, stat=teststat(model=loess))
pvalue <- sum(npboot$t > 0) / B
```



In this task we use the 2000 calculated T-values from the bootstrap and use these as the T-distribution and thus the share of T-values that do not follow the null-hypothesis will be the p-value. Since it is implied that if the fit of the local regression function has a decreasing pattern it will generate a negative T-value which is seen in the graph in section 1.2. Subsequently the test was formulated as follows:

$$H_o : T < 0$$
$$H_a : T \geq 0.$$

The p-value was calculated to 0.06 and we can't reject the null-hypothesis at a 5% significance level. We conclude H_o and that the lottery is random.

1.4

```
teststat_permutation<- function(data, B, stat) {
    n <- nrow(data)
    statistics <- rep(0, B)
    newdata <- data.frame(x=data$x, y=sample(data$y, n))

    for (b in 1:B) {
        statistics[b] <- stat(newdata)
        newdata$y <- sample(data$y, n)
    }
```

```
    sum(abs(statistics) >= abs(stat(data))) / B
}

set.seed(123456)
pvalue2 <- teststat_permutation(data, B, teststat(loess))
```

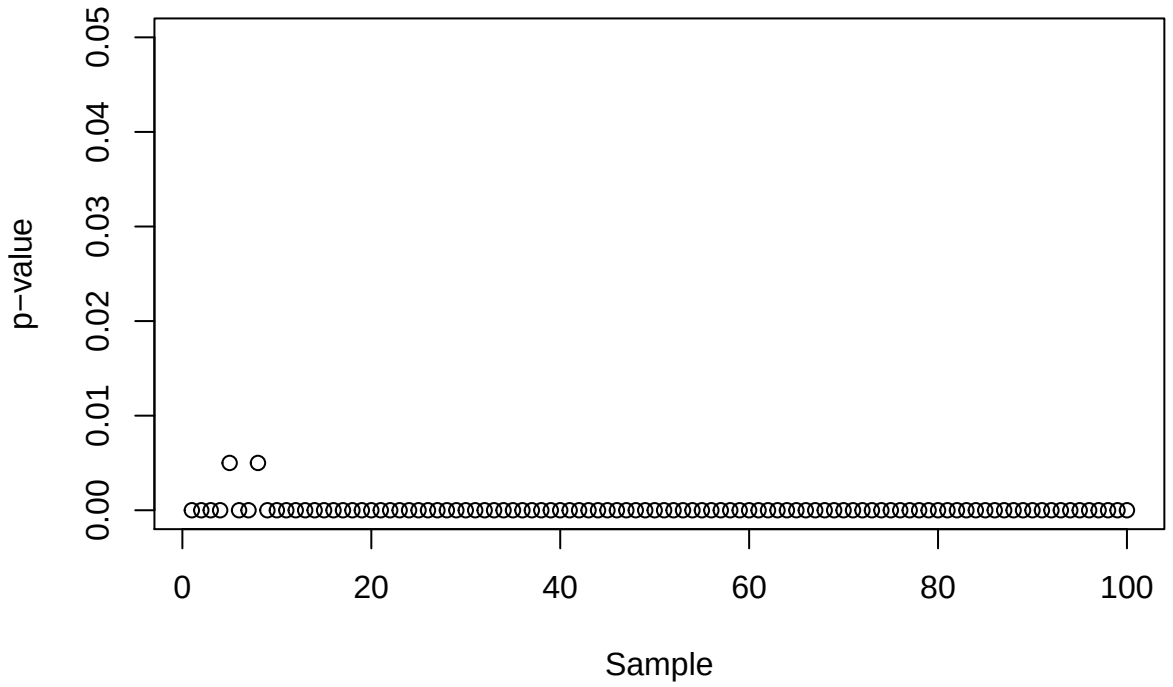
In permutation tests we compare the original T-statistic with T-statistics where the Draft No. are shuffled randomly. In this sense if these diverge much from the original T-statistic the lottery is random. If they are similar we will get a lower p-value since the quota will be smaller.

In this case the p-value is 0.09 and at a 5% significance level we conclude H_o in this test as well, and assert that the lottery is random.

1.5

To test that our test statistics is reliable in determining whether the lottery is random or not we used non-randomly generated data to calculate the power, i.e. 1 - Type-II errors (false negatives).

```
genranddata <- function(x, alpha) {
    data.frame(x=x, y=pmax(0, pmin(alpha * x + rnorm(length(x), mean=183, sd=10), 366)))
}

alphas <- seq(0.1, 10, by=0.1)
pvalues <- rep(0, length(alphas))

set.seed(123456)

for (i in 1:length(alphas)) {
    newdata <- genranddata(data$x, alphas[i])
    pvalues[i] <- teststat_permutation(newdata, 200, teststat(loess))
}

rejectionrate <- sum(pvalues <= 0.05) / length(pvalues)
```



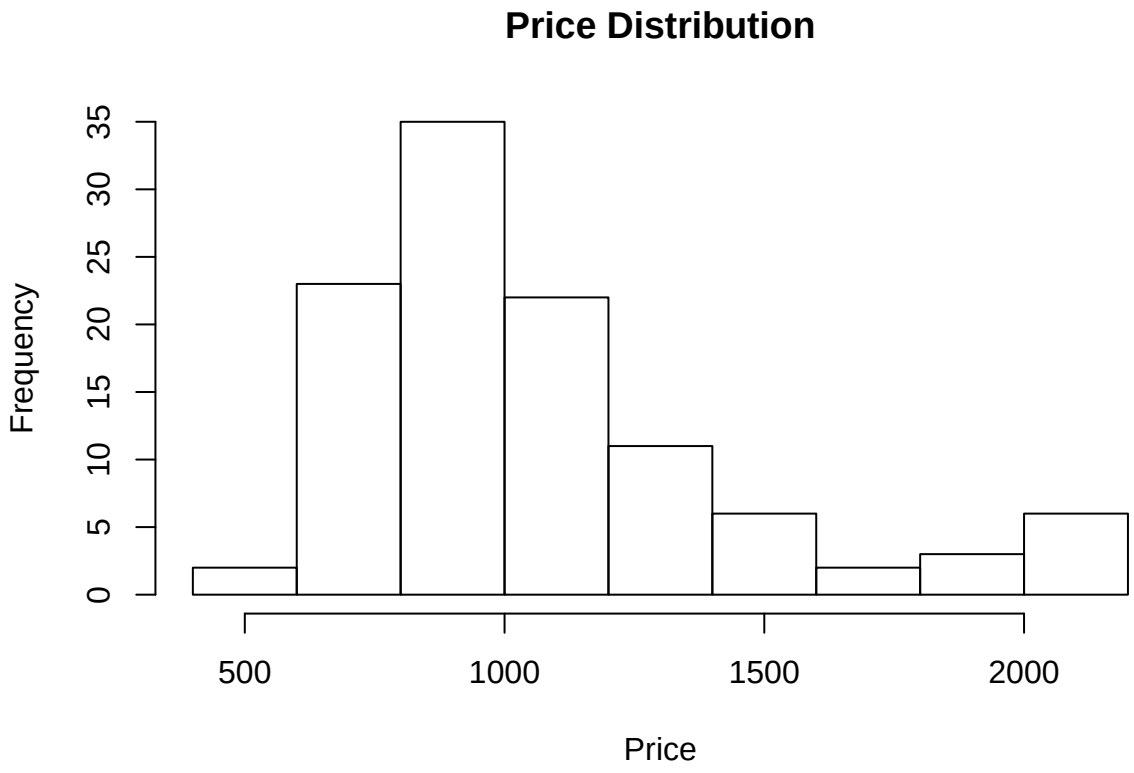
The rejection rate is 1 and since we reject H_o in every test the power will be 1 because we cannot have any Type-II errors when we reject everything. This is not strange since the test data is obviously non-random which gives us an indication that the test statistic is a good one for determining whether the data is randomly generated or not.

Question 2

In this task we are supposed to estimate the mean price of houses together with the variance using bootstrap and the jackknife method.

2.1

```
price <- read.csv("../data/prices1.csv", sep=";")
```



The mean price of the data set is 1080.47 and we think the histogram above looks like a Gamma distribution.

2.2

To calculate the bias corrected mean estimate we used

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

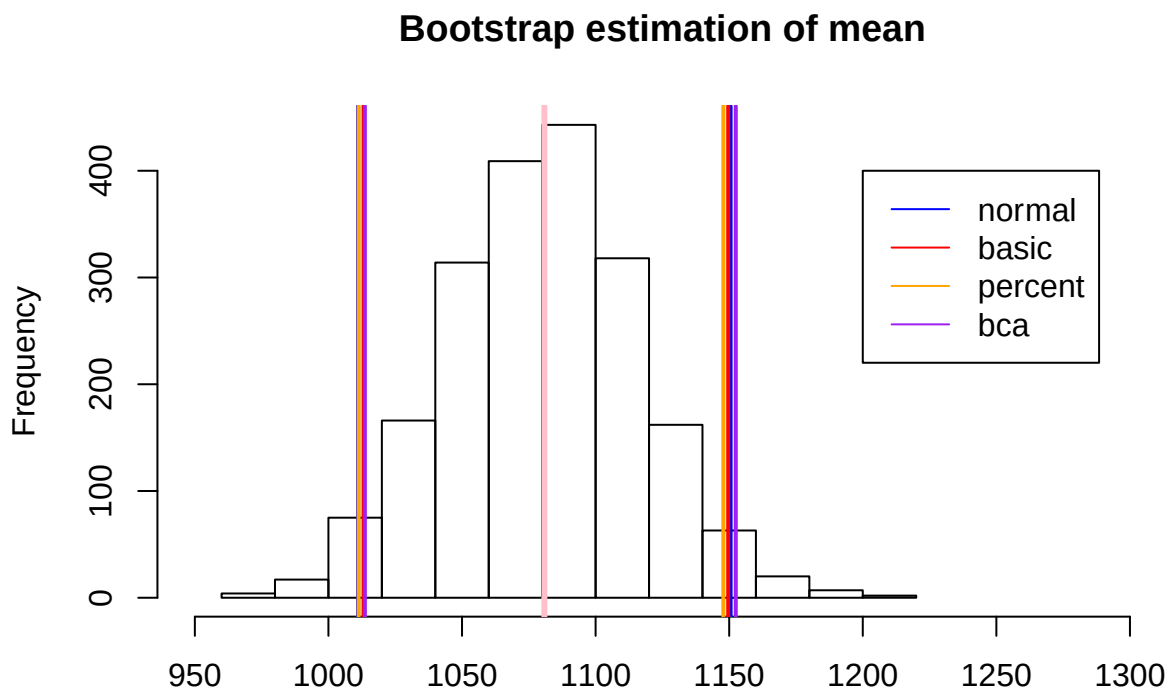
and to estimate the variance we used

$$\widehat{Var\left[T(\cdot)\right]} = \frac{1}{B-1}\sum_{i=1}^{B}\left(T(D_i^*) - \overline{T(D^*)}\right)^2.$$

D^* denotes the bootstrap samples while D is the actual sample given to us. Below is the result we got from the bootstrap sampling.

```
bootmean <- function(data,ind){
  data <- data[ind]
  mean(data)
}

B <- 2000

estmean <- boot(data = price$Price,bootmean, R = B)

bias_corrected_estmean <- 2 * estmean$t0 - mean(estmean$t)

bias_correction <- mean(estmean$t- estmean$t0)

estvar <- sum((estmean$t - mean(estmean$t))^2) / (B - 1)

cibo <- boot.ci(estmean)
```

The bias corrected estimate of the mean is 1080.83, the bias correction, $\sum_{i=1}^{B}(T(D_i^*) - T(D))$, is -0.36, and the variance is estimated to 1264.3.



We can see from the histogram above that all the different estimates of the confidence interval provided by the boot-library are fairly similar.

2.3

We choose to set the number of jackknife samples to be equal to the number of samples in our data set since then we can estimate the variance by using

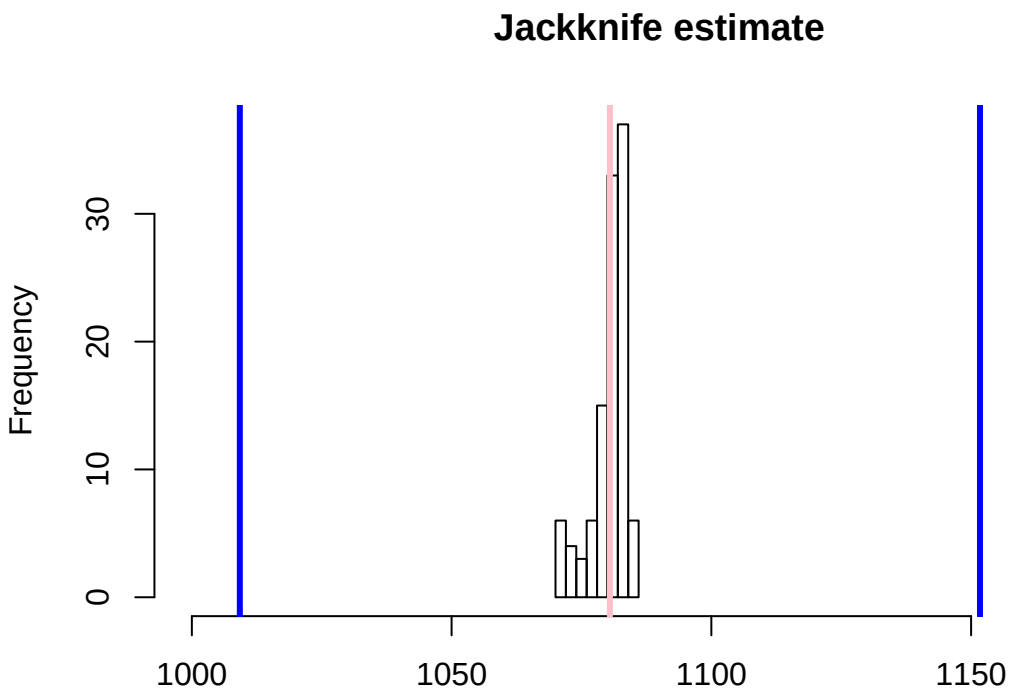
$$\widehat{Var}[T(\cdot)] = \frac{1}{n(n-1)} \sum_{i=1}^{n} (T_i^* - J(T))^2$$

where,

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

```
jackknife <- function(data,B,tstat){
  stopifnot(B >= 0 && B <= length(data))

  est <- rep(1, times = B)

  for(i in 1:B){
    est[i] <- tstat(data[-i])
  }

  return(est)
}

jackest <- jackknife(price$Price, B=length(price$Price), tstat=mean)
jackestmean <- mean(jackest)

n <- length(price$Price)

tstar <- n * mean(price$Price) - (n - 1) * jackest
JT <- mean(tstar)
jackvar <- sum((tstar - JT)^2) / (n * (n - 1))

lowerci <- JT - 1.96 * sqrt(jackvar)
upperci <- JT + 1.96 * sqrt(jackvar)
```

We get the estimate of mean to be 1080.47 and variance estimate to 1320.91 which is similar to what we found previously with the bootstrap method.



As expected the sample does not vary particular much but since we are using the formula previously mentioned for calculating the variance we adjust for the dependence of the samples and we can see a wide confidence interval similar to the ones estimated by the bootstrap method.

2.4

Name	Lower	Mean	Upper
Normal	1011	1080	1151
Percent	1011	1080	1148
Basic	1013	1080	1150
BCa	1014	1080	1152
Jackknife	1009	1080	1152

The normal, percent, and the basic along with the Jackknife confidence intervals are moving in a similar range with only small differences in-between them. The only one sticking out is the BCa CI with overall higher estimations than the rest. The mean rounded is estimated to be the same for both the bootstrap and jackknife methods.