# SDA- Assignment 1

Jose Chacon (2699643)

## Exercise 4.1

### a) Graphical Exploration of the Sample

In the first exercise, we will graphically explore the sample to investigate its underlying distribution. The sample contains the birth weight of 189 observations. Our goal is to estimate a distribution, where the data could have originated from.

Each different graph reveals relevant insights into the underlying distribution as well as the sample's properties. We initialize our analysis by creating the following graphs: histogram, symplot, and box plot. We obtain the following insights from the plots:

- histogram: we see a bell-shaped histogram with a very dense middle part. It can also be described as fairly symmetric.
- symplot: we see that most of the points lie on the line except for the high-value observations, which fall under the median line.
- boxplot: we support the idea of symmetry in the data. We can also see one outlier at the lower part of the plot.

According to the systematic approach for searching for the underlying distribution and insights obtained before, we proceeded to test whether the sample comes from a normal distribution. We start by plotting the QQ plot, which strongly suggests normality in the sample. Furthermore, we complete our normality analysis by using the Shapiro-Wilk test at a confidence level of 95%. The p-value of the test was 0.43, so we cannot reject the hypothesis of normality. Taking into consideration that all graphs and supporting tests have indicated normality in the sample, we conclude that the normal distribution is a very strong candidate or even the best candidate for the underlying distribution of the sample.

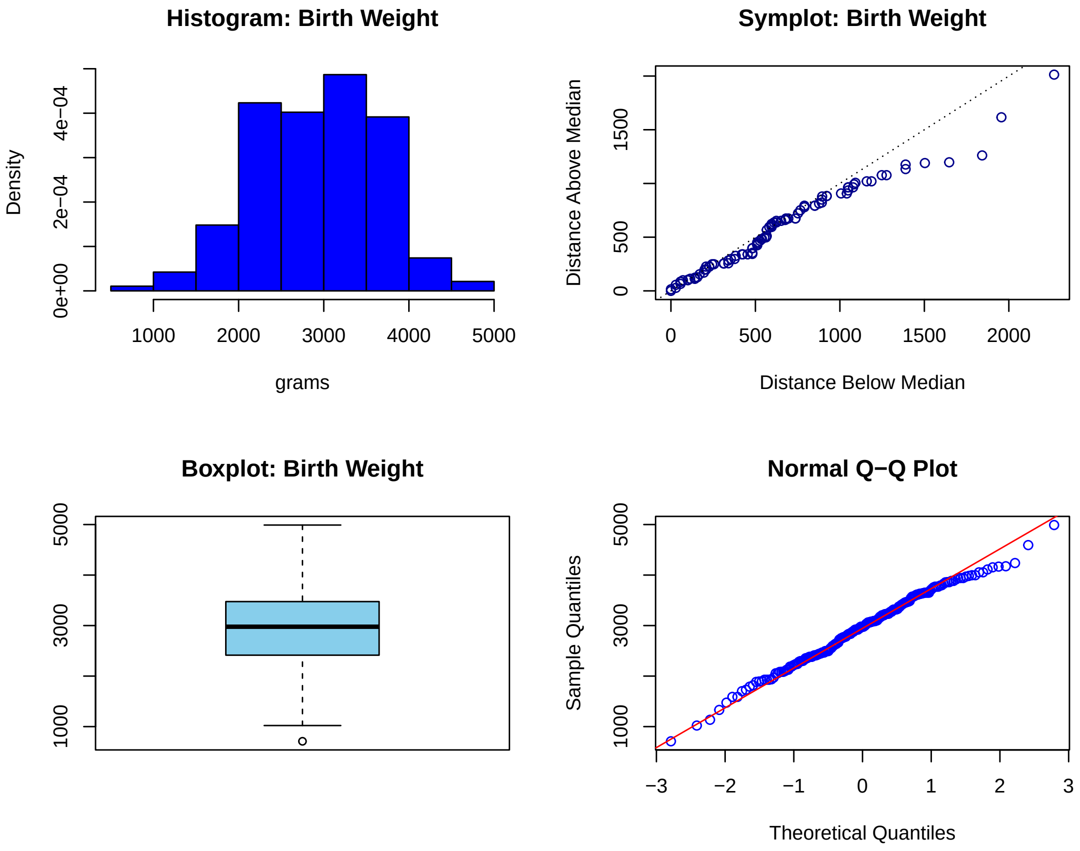

Figure 1: Graphical Exploration of the Sample

Furthermore, we investigated the possibility of a better representation of the distribution. We investigated many distributions by plotting their corresponding QQ plots against the sample. From these visualizations, we realize there are some other candidates worthy of consideration. The distributions are the following: logistic, chi-square (with a high degree of freedom value), and t-distribution (also with a high value for the degree of freedom). Unfortunately, the tests performed to corroborate any of these distributions failed. Therefore, we concluded to use the normal distribution as the underlying distribution for this sample, but these three other candidate distributions can be considered for specific procedures if needed. The QQ-plots for the logistic, chi-square, and t-distribution (both with a high value for the df) will be shown in the appendix

## b) Bootstrap methods: Standard deviation of the 5%-quantile

The goal of this section is to estimate the 5%-quantile using the bootstrap method. First, we have estimated the 5%-quantile of the sample, which is equal to 1801. Secondly, we performed two types of bootstrap methods: the empirical and the parametric (using the normal distribution with the sample mean and sample standard deviation). We chose the normal distribution due to the findings in subsection a).

For both bootstrap methods, we used B=2000 to ensure convergence to the true values. For the empirical method, we obtained a standard deviation of the 5%-quantile of 115.96, while using the parametric method we obtained a standard deviation of 111.46. This means that the parametric bootstrap method yielded a better estimate due to less variability in the sample. The histograms of both bootstrap method samples are shown below.

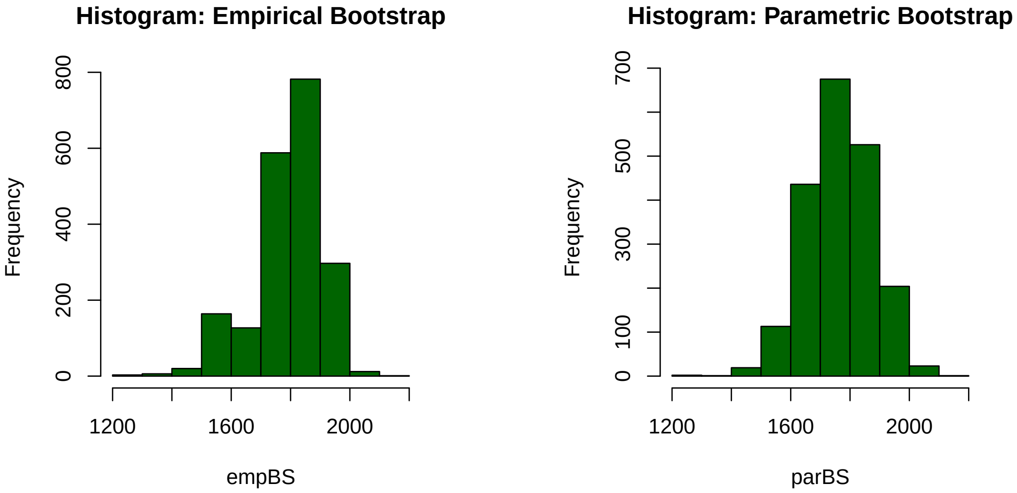

Figure 2: Histograms for the different bootstrap methods

## c) Parametric Bootstrap based on Log-normal Distributions

In this section, we have also computed the parametric bootstrap estimate for the standard deviation of the 5%-quantile. However, in this case, we have used the log-normal distribution as the assumed underlying distribution instead of the normal distribution. To obtain the parameters for the bootstrap method, we first logarithmized the sample and computed the mean and standard deviation of that transformed sample, which are the parameters.

In this case, we obtained an average standard deviation for the 5%-quantile of 74.60. This estimate is smaller than the previous parametric bootstrap estimate and therefore, a better estimate for the true standard deviation for the 5%-quantile. The reason for this better approximation can be that the assumed distribution in the first parametric bootstrap estimate was wrong. This would mean that Error type 1 (choosing a different underlying distribution as the ”real” one) is present, which usually has a large impact on the results. Additionally, we know that the log-normal distribution is less sensitive to outliers than the normal distribution. Therefore, this could be translated into a better estimation of the standard deviation.

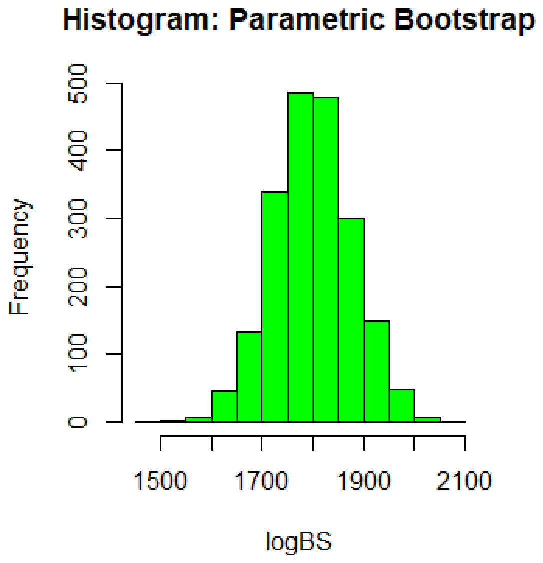

Figure 3: Histograms for the different bootstrap methods

### d) One-line Code for the bootstrap method

Finally, we were tasked to code the parametric bootstrap estimate of the 5%-quantile calculated in section c) in one line of code. Just as before, we have assumed that the underlying distribution is log-normal. The result is the following:

```
sdEstimator = sd(replicate(2000, quantile(rlnorm(length(sample41),
                mean=mean(log(sample41)), sd=sd(log(sample41))), prob=0.05)))
```

## Exercise 4.2

For this exercise, we are given 3 samples inside the file *thrombolobulin.txt*. We are asked to create some confidence intervals for some specific statistic (with a certain level of confidence). We have created a general recipe to compute the confidence interval using the bootstrap method. The recipe is as follows:

1. Generate B bootstrap samples of the same size as the original sample.
2. Compute the statistic of each of the bootstrap samples and save this vector as *tstar*.
3. Save the sample median in a vector called *tn*.
4. Compute *zstar* by the formula: *tstar* - *tn*
5. Compute the limits of the interval with the following formula:
   lowerCI = tn - quantile(zstar, 1-$\alpha$/2)
   upperCI = tn - quantile(zstar, $\alpha$/2)
6. CI = [lowerCI, upperCI]

Notes: there could be some changes depending on the purpose and calculation form of the confidence intervals. On another note, we have mainly used B=2000 to avoid any type 2 errors.

### a) Two-sided 95%-Bootstrap Confidence Interval for the median

The first task was to determine the confidence interval (at a confidence level of 95%) for the median of the underlying distribution of SDRP. We have used the recipe above to compute this CI. In this case, we use the median as the statistic and 5% as our alpha (equivalent to 95% confidence level). The result of this procedure is the 95%-bootstrap confidence interval for the median of the SDRP sample, which is [43.5, 65.5]. This makes sense taking into consideration that the sample median is 49.5

## b) Two-sided 95%-Bootstrap Confidence Interval for the mean

This part of the assignment was very similar to the previous with the exception that now we want to calculate the confidence interval for the mean of the sample. We again are looking for a 95%-bootstrap confidence interval. Therefore, we performed a bootstrap method for 2000 sample estimates and calculated the mean of each sample. In this case, we obtained a confidence interval of [48.1, 72.6]. This CI is also in alignment with our sample's mean, which equals 61.93.

## c) Best location estimator

When we compare the confidence intervals for the mean and the median, we can conclude that the median estimate is a better estimator of location than the mean estimate. This is due to several reasons. First, the histogram is heavily skewed to the left and it contains some very large values. This causes a displacement of the mean, which is to the left of the histogram (instead of the center). On the other hand, the median is located in a more central position in the histogram, so it can better describe the location. The histogram will be shown at the end of this section.

When we compare the standard deviation of the mean bootstrap sample (containing the mean of the bootstrap samples) with the median bootstrap sample, we support even more our choice of median as a better estimation. As it is clear that the median sample has less variation than the mean bootstrap sample.

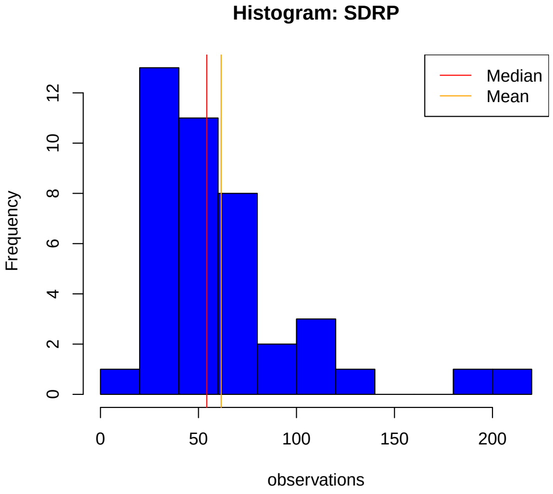

Figure 4: Histogram of the sample SDRP

## d) Two-sided 90%-Bootstrap Confidence Interval for the difference in the median between groups SDRP and CTRP

There are some big differences between the computation of this confidence interval with the ones computed before. First, in this case, we need to re-sample using bootstrap from 2 samples simultaneously. This is because we are looking for the confidence interval for the difference in the median between the SDRP and CTRP samples. Therefore, we have created a function that helps us re-sample using bootstrap from both samples simultaneously. What this formula return is equal to the *tstar* described in the recipe for CI. The formula in R code is the following:

```
BS_diffMedian = function(sample1, sample2, B=1000){
  BS_sample = numeric()
  for(i in 1:B){
    median1 = median(sample(sample1, size=max(sample1, sample2), replace=TRUE))
    median2 = median(sample(sample2, size=max(sample1, sample2), replace=TRUE))
    BS_sample = c(BS_sample, median2 - median1)
  }
  return (BS_sample)
}
```

From the formula, we can see that, in each iteration, we re-sample twice for each of the samples. We directly calculate the statistic of each re-sample, in this case, the median. We save the difference of the statistic on a pre-defined vector and return the vector with B elements, each element is the difference of median of the bootstrap samples of the parameter samples. As previously stated, the return data structure contains the value of *tstar*.

After this procedure, we can proceed as in the recipe to compute the CI for the difference in the median of the SDRP and CTRP samples. In this case, we use a 90% confidence level instead of the 95% used in previous examples. The results yielded a CI equal to [-27, 0.5]. We believe this is also in alignment with what we saw from the original samples (a median difference of -13).

We can conclude from this interval for the difference between the groups SDRP and CTRP that the CTRP group has lower $\beta$-thromboglobulin levels. However, because we can find the value 0 is the interval, we have to confirm the doubt about the existence of a systematic difference between the 2 groups.

## Exercise 4.3

In this exercise, we are given two sets of samples that come from the experiments performed by Michelson in 1879 and 1882. The main objective of this exercise is to test if the samples originate from a normal distribution. To test this hypothesis we use the modified Kolmogorov-Smirnov test.

The modified Kolmogorov-Smirnov test statistic for a sample of size $n$ is given by:

$$D_n = \sup_x |F_n(x) - \Phi\left(\frac{x-\overline{X}}{S}\right)| = \max_i \left(\max\left(\left|\frac{i-1}{n} - \Phi\left(\frac{x_i-\overline{X}}{S}\right)\right|, \left|\frac{i}{n} - \Phi\left(\frac{X_i-\overline{X}}{S}\right)\right|\right)\right)$$

where $F_n(x)$ is the empirical distribution function, $\Phi$ is the standard normal cumulative distribution function, $\overline{X}$ is the sample mean, $S$ is the sample standard deviation, $i$ is an index that ranges from 1 to $n$, and $x_i$ and $X_i$ are the $i$-th order statistics of the sample.

### a) Independent from the location and scale parameters

From the theory, we know that if the underlying distribution is continuous, the statistic becomes non-parametric. Therefore, it becomes independent of the location and scale location. In other words, the modified statistics are invariant to any strictly increasing transformation of the sample, such as the multiplication or addition of a constant. However, the test works because it uses the rank (or other statistics) of the sample data to compute the test.

### b) Testing the Null Hypothesis with the adjusted Kolmogorov-Smirnov test statistic

The second step of this analysis was to test the null hypothesis with the adjusted Kolmogorov-Smirnov Test (mentioned above) at a 5% significance level. We started by combining both samples into one. Then we measured the modified statistic for the sample, for which the result is 0.067 (above the 5% significance level). We used this to compare it with the results from the bootstrap method.

First, we used the empirical bootstrap method to compute 1000 samples from the original sample. We computed the modified test statistic of each of these samples and saved them into a new vector. The average value of the test statistic in the vector is 0.09. We used this new vector with the test statistics to compute its corresponding histogram. (Due to lack of importance, we will not present this histogram).

However, the empirical bootstrap method is not the best choice to test the sample under this null hypothesis. This is because we must ensure that the bootstrap sample also follows the normal distribution. Therefore, we used the parametric bootstrap method. We created 1000 random samples from a normal distribution with mean, standard deviation, and size of the original sample. Then we computed the statistic for each sample and created a new vector containing these 1000 statistics. The average value of the test statistic using the parametric bootstrap is 0.056. We then computed the p-value as the proportion of the bootstrap samples that have a greater than or equal to the observed statistic (Dn=0.067) and the result is 0.187, which is greater than the significance level. Therefore, we do not reject for normality.

We created a histogram from the vector with the bootstrap statistics to visualize the solution. The histogram can be found in the following figure:

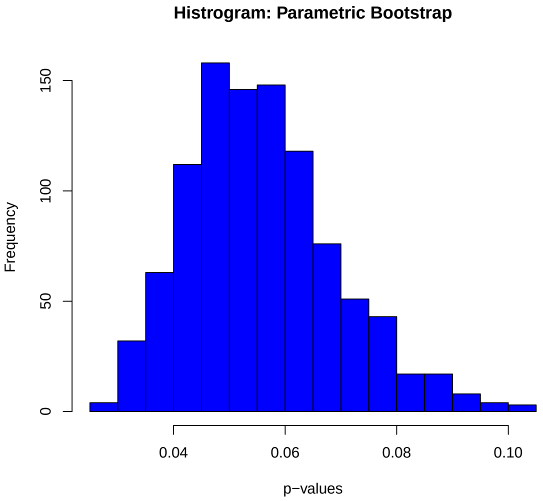

Figure 5: Histogram of the p-value

## c) Comparison of p-values

When comparing the p values found in b from those obtained using the output of KS test when one uses as input the estimated mean and standard deviation. There exists a difference between these two p-values. The main reason behind this difference is that the KS test assumes that the mean and standard deviation are known, and uses these values to ultimately compute the p-values (through the test statistics). On the other hand, when using the bootstrap method we are using the sample mean and sample deviation to generate random samples. We then used these samples to generate a distribution of the test statistic. Due to the randomness in the samples (also the uncertainty with the population mean and standard deviation), this can lead to different p-values.

# Appendix

## 4.1 Plots

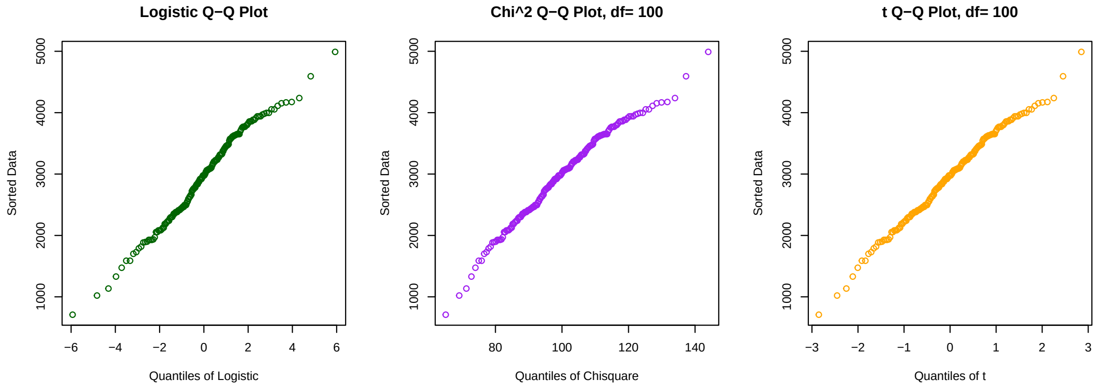

Figure 6: QQ Plots for the Logisitc, Chisquare and T distribution

## 4.1 Code

```
#a)
setwd("C:\\Users\\josec\\OneDrive\\Documentos\\VU\\Statisitcal Data Analysis\\Assignment 4")
source("functions_Ch3.txt")
source("functions_Ch5.txt")
sample41 = scan("birthweight.txt")

par(mfrow=c(2,2))
hist(sample41, freq=FALSE, main='Histogram: Birth Weight', xlab='grams', col='blue')
symplot(sample41, main="Symplot: Birth Weight", col='darkblue')
boxplot(sample41, main='Boxplot: Birth Weight', col='skyblue' )
qqnorm(sample41, col='blue')
qqline(sample41, col='red')

par(mfrow=c(1,1),pty='s')
qqnorm(sample41)
qqline(sample41, col='red')
shapiro.test(sample41) #p-value=0.43 -> cannot reject normality

par(mfrow=c(1,3),pty='s')
qqlogis(sample41, col='darkgreen') #good approx.
qqchisq(sample41, df=100, col='purple') #also very good
qqt(sample41, df=100, col='orange') #best candidate so far

qqcauchy(sample41)
qqexp(sample41)
qqunif(sample41)
qqlnorm(sample41)
qqlaplace(sample41)

ks.test(sample41, "plogis", 0, 1)
ks.test(sample41, "pt", df=15)
ks.test(sample41, "pchisq", df=100)
ks.test(sample41, "plogis")
ks.test(sample41, "pnorm", exact=FALSE)
```

```
#b)
quant5 = quantile(sample41, prob=0.05) #1801
statistic = function(x){ quantile(x, prob=0.05)}

empBS = bootstrap(sample41, statistic = statistic, B=2000)
empBS.sd = sd(empBS) #115.96
empBS.mean =mean(empBS) #1791.27

set.seed(202020)
parBS = replicate(2000, statistic(rnorm(length(sample41), mean=sample.mean,
sd=sample.sd)))
parBS.sd = sd(parBS) #111.46
parBS.mean = mean(parBS) #1762.71

#c)
log_sample = log(sample41)
log.mean= mean(log_sample)
log.sd = sd(log_sample)

set.seed(202020)
logBS = replicate(2000, statistic(rlnorm(length(sample41), mean=log.mean,
sd=log.sd)))
logBS.mean = mean(logBS) #1799.85
logBS.sd = sd(logBS) #77.50

#d)
set.seed(202020)
sd(replicate(2000, quantile(rlnorm(length(sample41), mean=mean(log(sample41)),
sd=sd(log(sample41))), prob=0.05)))
```

### 4.2 Code

```
setwd("C:\\Users\\josec\\OneDrive\\Documentos\\VU\\Statisitcal Data Analysis\\Assignment 4")
source("functions_Ch5.txt")
source("thromboglobulin.txt")
sampleSDRP = thromboglobulin$SDRP
samplePRRP = thromboglobulin$PRRP
sampleCTRP = thromboglobulin$CTRP

hist(sampleSDRP, breaks=10, main=’Histogram: SDRP’, xlab=’observations’, col=’blue’)
abline(v=tn, col=’red’)
abline(v=tn2, col=’orange’)
legend("topright", legend = c("Median", "Mean"), col = c("red", "orange"), lty = 1)

#a) determine a two-sided 95%-CI for the median of SDRP
tstar = bootstrap(sampleSDRP, median, B=2000)
tn = median(samplePRRP)
zstar = tstar - tn
CI_median = c(tn-quantile(zstar,0.975), tn-quantile(zstar, 0.025)) #[43.5, 65.5]

#b)
tstar2 = bootstrap(sampleSDRP, mean, B=2000)
tn2 = mean(samplePRRP)
zstar2 = tstar2 - tn2
```

```
CI_mean = c(tn2-quantile(zstar2,0.975), tn2-quantile(zstar2, 0.025)) #[48.1, 72.6]

#c)
CI_median[2] - CI_median[1] #22
CI_mean[2]- CI_mean[1] #22
var(tstar) #median: 31.63
var(tstar2) #mean: 39.97

#d) determine a 90%-CI for diff median between SDRP and CTRP
BS_diffMedian = function(sample1, sample2, B=1000){
  BS_sample = numeric()
  for(i in 1:B){
    median1 = median(sample(sample1, size=length(sample1), replace=TRUE))
    median2 = median(sample(sample2, size=length(sample2), replace=TRUE))
    BS_sample = c(BS_sample, median2 - median1)
  }
  return (BS_sample)
}

tstar3 = BS_diffMedian(sampleCTRP, sampleSDRP)
tn3 = median(sampleSDRP)-median(sampleCTRP)
zstar3 = tstar3 - tn3

CI_diff = c(tn3-quantile(zstar3,0.95), tn3-quantile(zstar3, 0.05)) #[-27, 0.5]
```

### 4.3 Code

```
setwd("C:\\Users\\josec\\OneDrive\\Documentos\\VU\\Statisitcal Data Analysis\\Assignment 4")
source("functions_Ch5.txt")
light= source("light.txt")

sample43 = c(light$value$'1879', light$value$'1882')

Dn = ks.test(sample43, "pnorm", mean = mean(sample43), sd = sd(sample43), alternative = "two.si

D = function(sample){
  ks.test(sample, "pnorm", mean = mean(sample), sd = sd(sample), alternative='two.sided')$stati
}

#empirical BS
BS = bootstrap(sample43, D, B=1000)
mean(BS) #0.09
hist(BS, col='blue', xlab='p-values', main='Histrogram: Empirical Bootstrap')
abline(v= Dn, col='red')

#parametric BS
set.seed(20202020)
BS2= replicate(1000,D(rnorm(length(sample43), mean = mean(sample43), sd=sd(sample43))))
mean(BS2) #0.056
hist(BS2, col='blue', xlab='p-values', main='Histrogram: Parametric Bootstrap')

p_val = mean(BS2 >= Dn) #0.187
```