Assignment 2

Tanjina Islam, Miguel Morales Expósito and Carlos Perales Liñan, group 12

28 February 2018

Exercise 1

Task 1

We load the data from the data source.

```
Data <- read.table("telephone.txt",header = TRUE);
bills = Data$Bills;
```

We define our t statistics which is going to be the median.

```
t=median(bills);
```

We define the function for the bootstrap test for different lambda values.

```
bootstrap_exp<-function(n,lambda){
  B=1000;
  tstar=numeric(B);
  for(i in 1:B){
    xstar = rexp(n,lambda); #Exercise says for lambda = [0.01,0.1]
    tstar[i] = median(xstar);
  }
  return(tstar)
}
```

We define 200 hundred values for lambda and perform the bootstrap test on each of them.

```
lambda_values=seq(0.01,0.1,length=200)
p_array=numeric(length(lambda_values))
n=length(bills);
B=1000;
i=1
for(lambda in lambda_values){
  tstar=bootstrap_exp(n,lambda);
  pl=sum(tstar<t)/B;
  pr=sum(tstar>t)/B;
  p=2*min(pl,pr);
  p_array[i]=p;
  i=i+1;
}
```

We take the values where p is not rejected

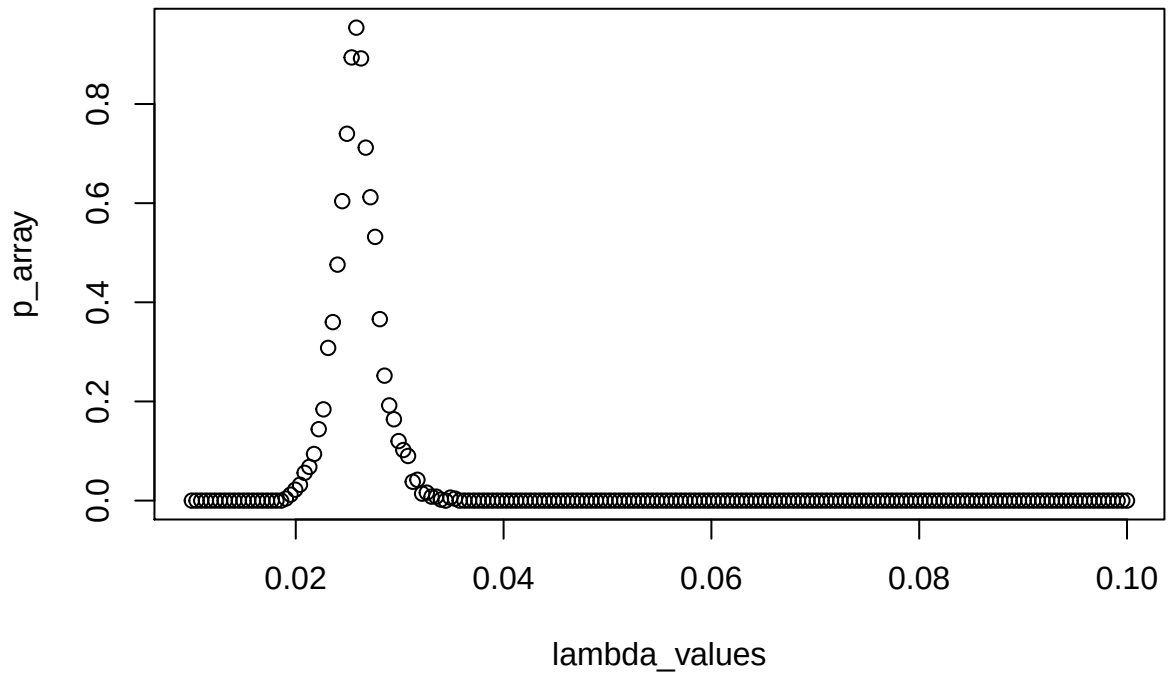
```
p_not_reject=p_array[p_array>0.05]
```

We get the corresponding lambda values for those p-values

```
indexes = match(p_not_reject,p_array);
lambda_not_reject=numeric(length(indexes));
count=1
for(i in indexes){
  lambda_not_reject[count]=lambda_values[i];
  count=count+1;
}
```

We cannot reject that the data stems from a exponential distribution with lambda values: 0.021, 0.021, 0.022, 0.022, 0.023, 0.023, 0.024, 0.024, 0.024, 0.025, 0.025, 0.026, 0.026, 0.027, 0.027, 0.028, 0.028, 0.029, 0.029, 0.029, 0.03, 0.03, 0.031

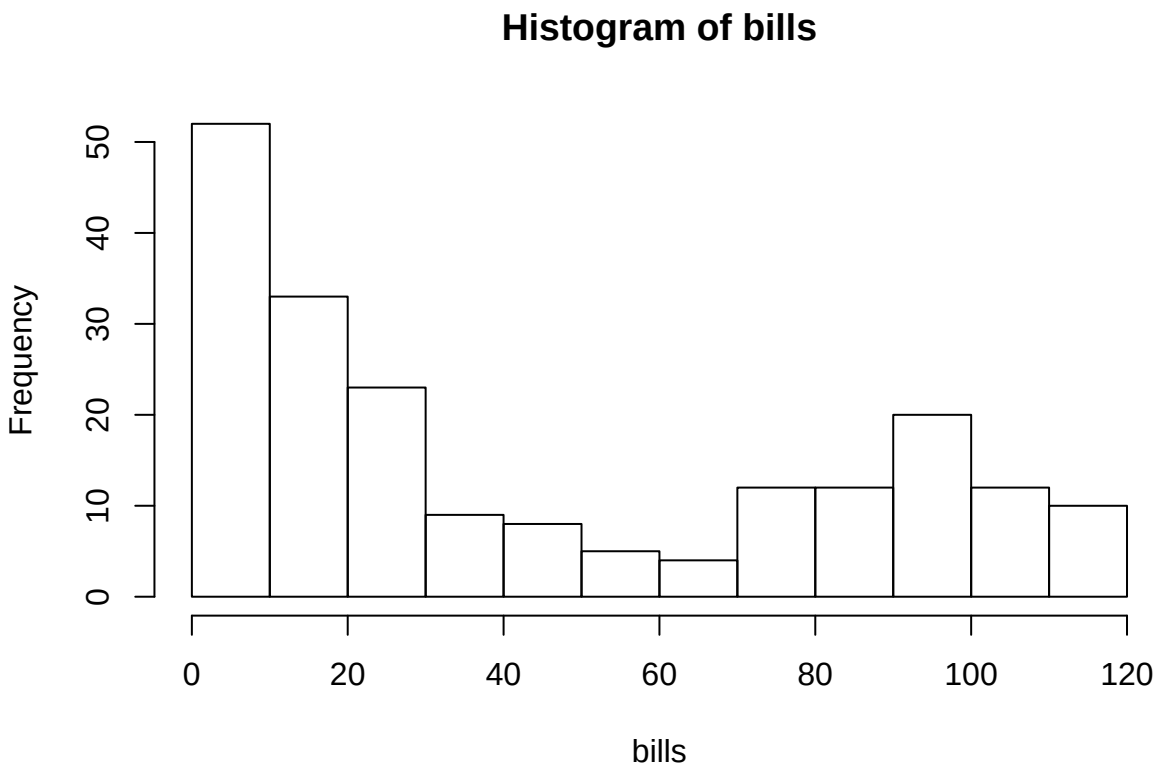
```
plot(lambda_values,p_array)
```



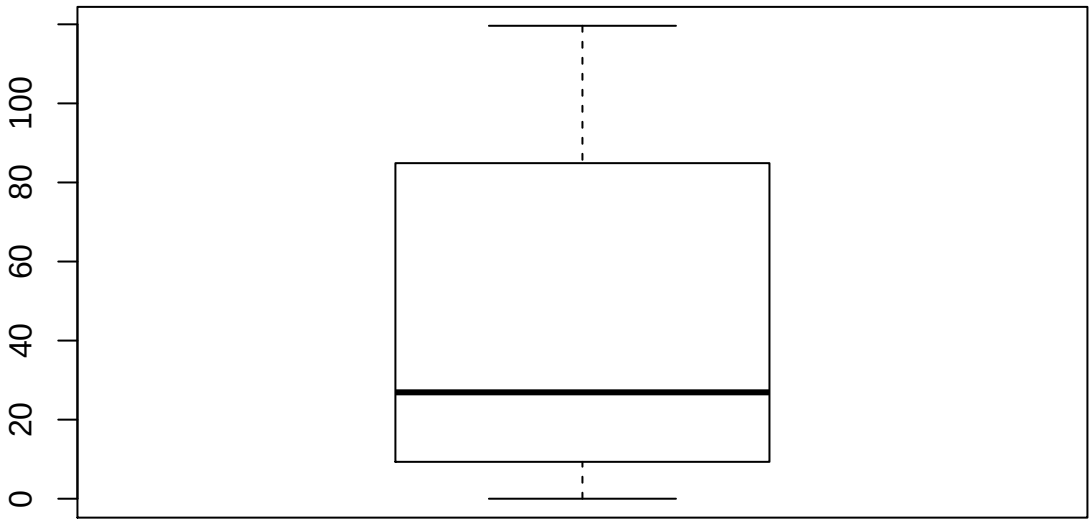
Looking at the graph and the previous lambda values, we can estimate that the data stems from a exponential distribution with lambda values in the range [0.021,0.031].

Task 2

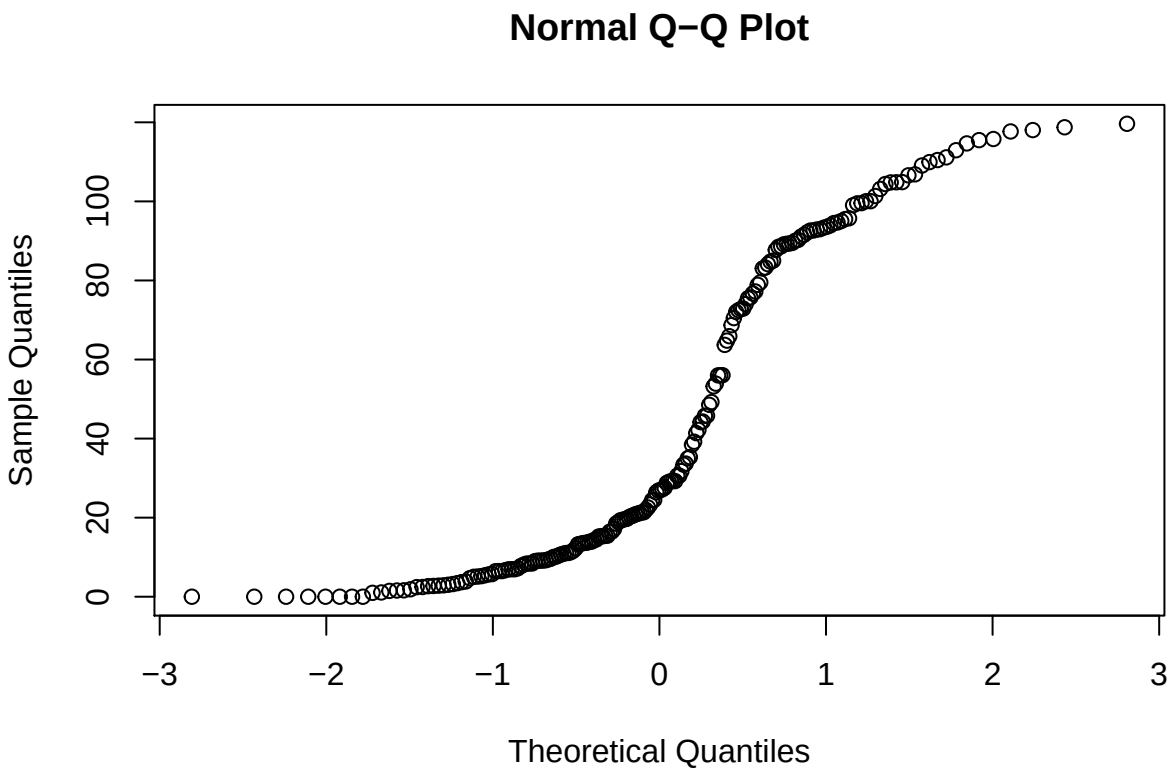
```
hist(bills)
```




```
boxplot(bills)
```




```
qqnorm(bills)
```



Considering that the median of the data is 26.91 the marketing manager could focus the marketing campaign saying that half of their customers spend less than 26.91 euros every month.

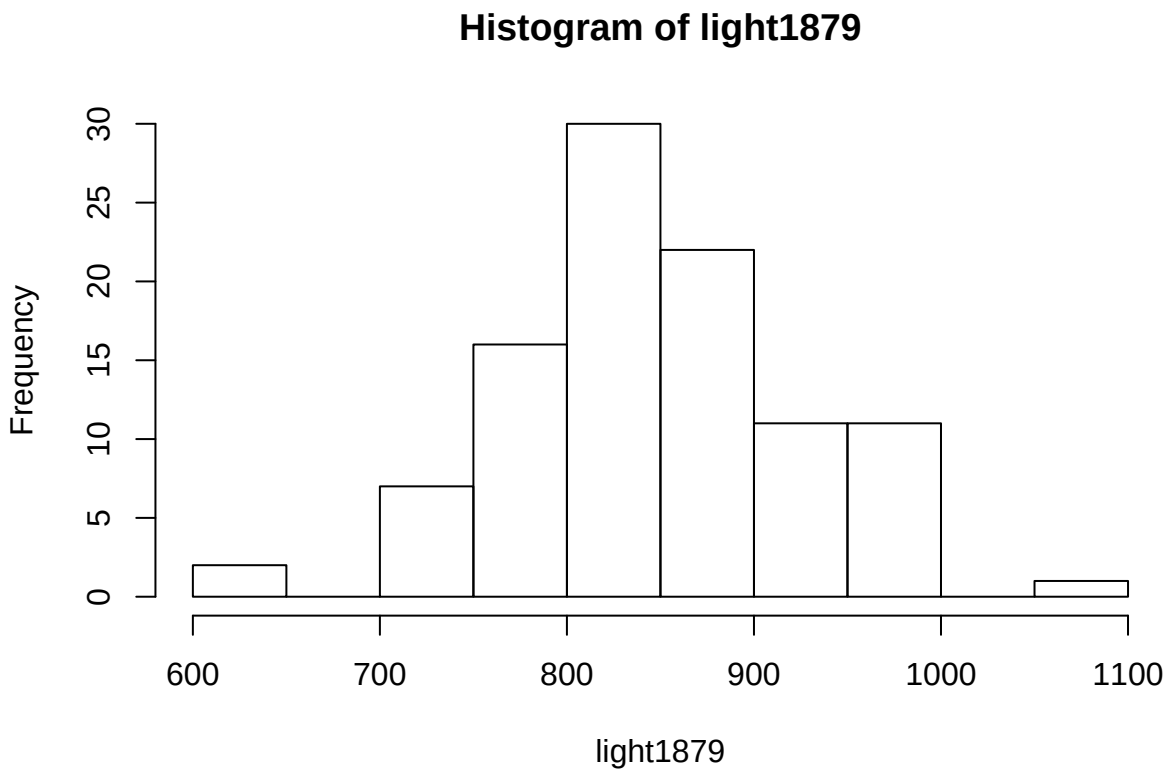
Exercise 2

We load the data and normalize it so the data represents the same from the different data sources

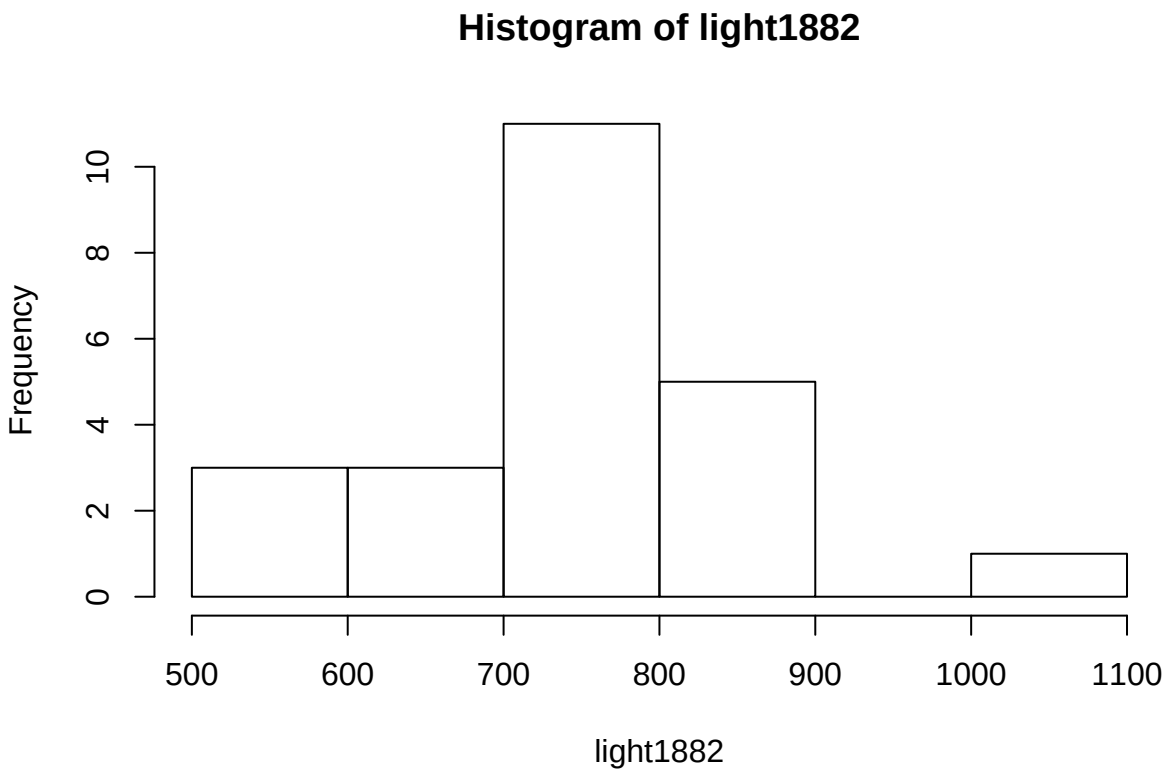
```
light1879 = scan("light1879.txt");
light1882 = scan("light1882.txt")
light_time = scan("light.txt")

light_real_time = ((light_time/1000)+24.8)/1000000
light_speed = (7.442/light_real_time)-299000
```

Taks 1

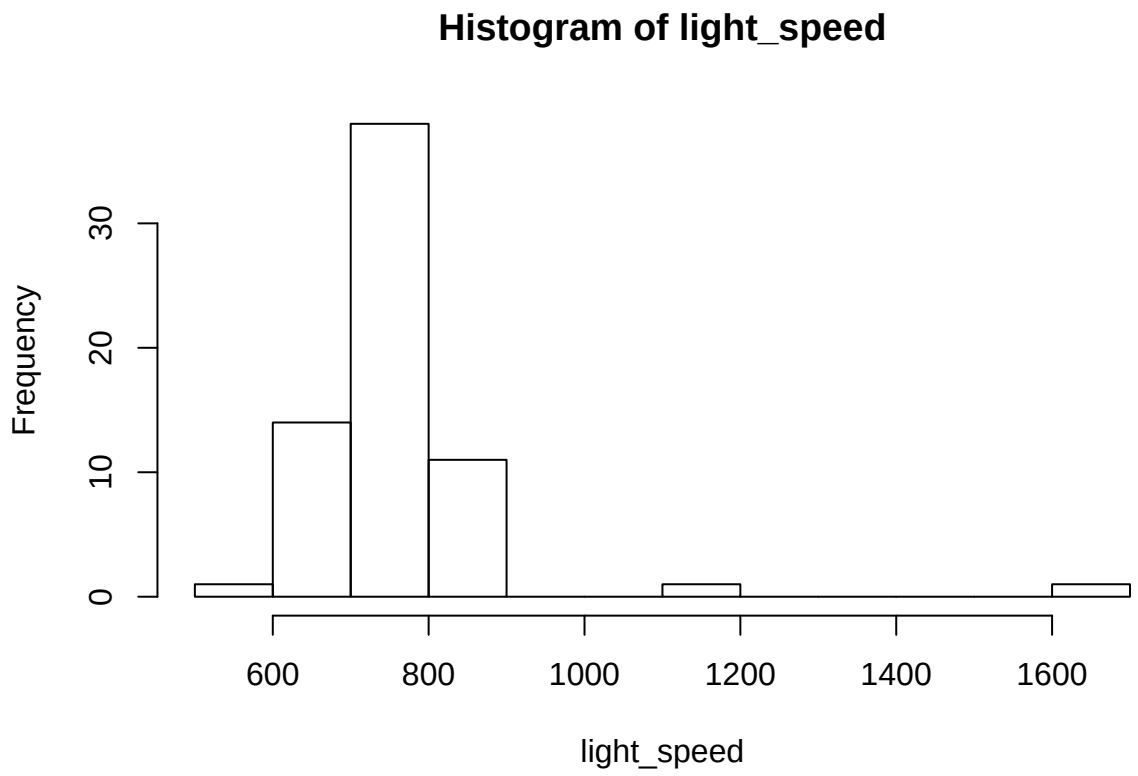
```
hist(light1879);
```



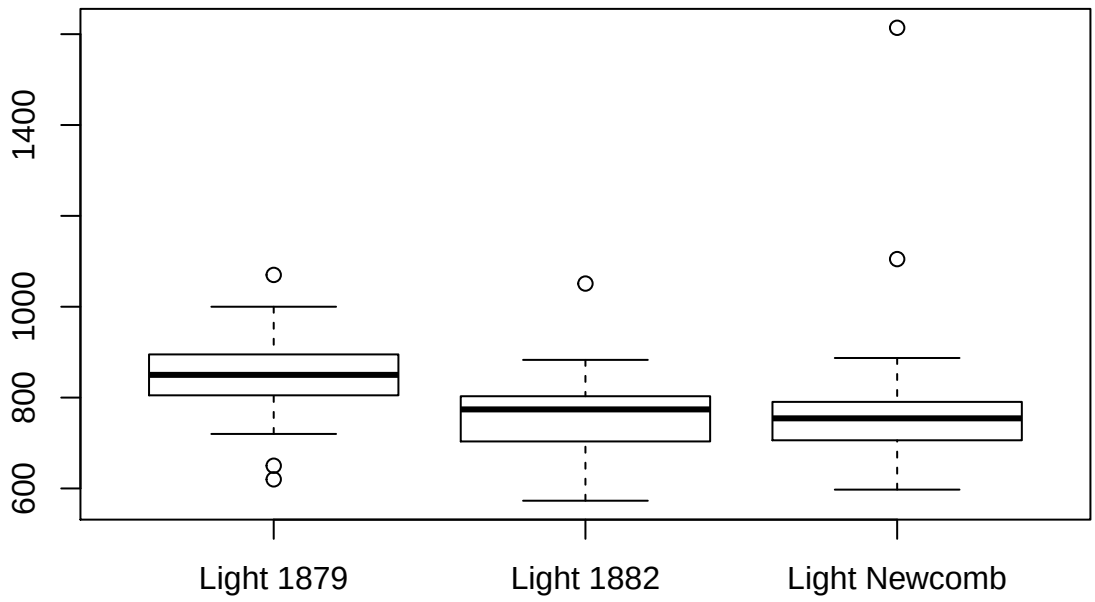

```
hist(light1882);
```




```
hist(light_speed);
```




```
boxplot(light1879,light1882,light_speed,names=c("Light 1879","Light 1882","Light Newcomb"))
```



Comparing the histograms and boxplots we can see that the distribution of the light1882 and light_speed are quite similar. Meanwhile, the light1879 the boxplot seems shifted upwards which means that the values are bigger.

If we consider the median values then we get for light1879 is 850, for 1882 is 774 and light_speed is 754.3. We see again that light1882 and light_speed medians are close to each other.

Task 2

We define a function to perform a bootstrap test

```
bootstrap<-function(data,operation="mean"){
  B=1000;
  tstar=numeric(B);

  for(i in 1:B){
    xstar = sample(data,replace = TRUE);

    if(operation=="mean"){
      tstar[i] = mean(xstar);
    }else if(operation=="median"){
      tstar[i] = median(xstar);
    }
```

```
  }
  return(tstar);
}
```

We define a function to get the confidence intervals

```
get_confidence_intervals<-function(data,operation="mean"){
  if(operation=="mean"){
    t=mean(data);
  }else if(operation=="median"){
    t=median(data);
  }
  tstar = bootstrap(data,operation);
  tstar25 = quantile(tstar,0.025);
  tstar975 = quantile(tstar,0.975);
  return(c(2*t-tstar975,2*t-tstar25));
}
```

Light 1879

```
confidence_mean = get_confidence_intervals(light1879);
confidence_mean = confidence_mean+299000; #We add 299000 to get the actual speed
```

The confidence interval for the mean is [299836,299867]

```
confidence_median = get_confidence_intervals(light1879,"median");
confidence_median = confidence_median+299000;
```

The confidence interval for the median is [299830,299860]

Light 1882

```
confidence_mean = get_confidence_intervals(light1882);
confidence_mean = confidence_mean+299000; #We add 299000 to get the actual speed
```

The confidence interval for the mean is [299710,299797]

```
confidence_median = get_confidence_intervals(light1882,"median");
confidence_median = confidence_median+299000;
```

The confidence interval for the median is [299751,299825]

Light Newcomb

```
confidence_mean = get_confidence_intervals(light_speed);
confidence_mean = confidence_mean+299000; #We add 299000 to get the actual speed
```

The confidence interval for the mean is [299730,299790]

```
confidence_median = get_confidence_intervals(light_speed,"median");
confidence_median = confidence_median+299000;
```

The confidence interval for the median is [299742,299766]

Task 3

Looking at the confidence intervals of light1882 and light Newcomb, we can find that they are similar and more or less are in the same range. Meanwhile, the confidence interval of 1879 is not in the same range as the other two for the mean and for the median. The intersection of the intervals of 1879 and the others is empty. Then the experiments of 1879 maybe are not accurate enough or at least they differ from the other two experiments.

Task 4

The speed of light according to https://physics.nist.gov/cgi-bin/cuu/Value?c is 299792.458.

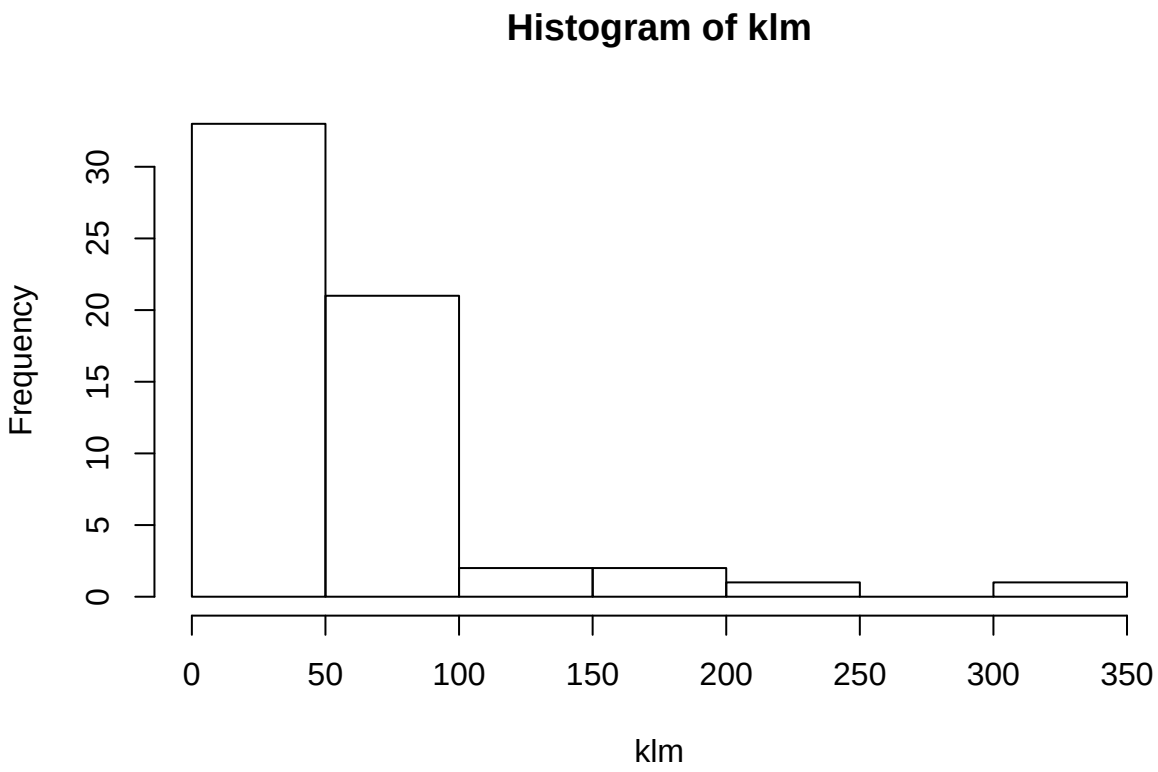
Checking the intervals in task2, we can see that the value of the speed of light only falls in the interval for the mean and median of the speed light of 1882.

Exercise 3

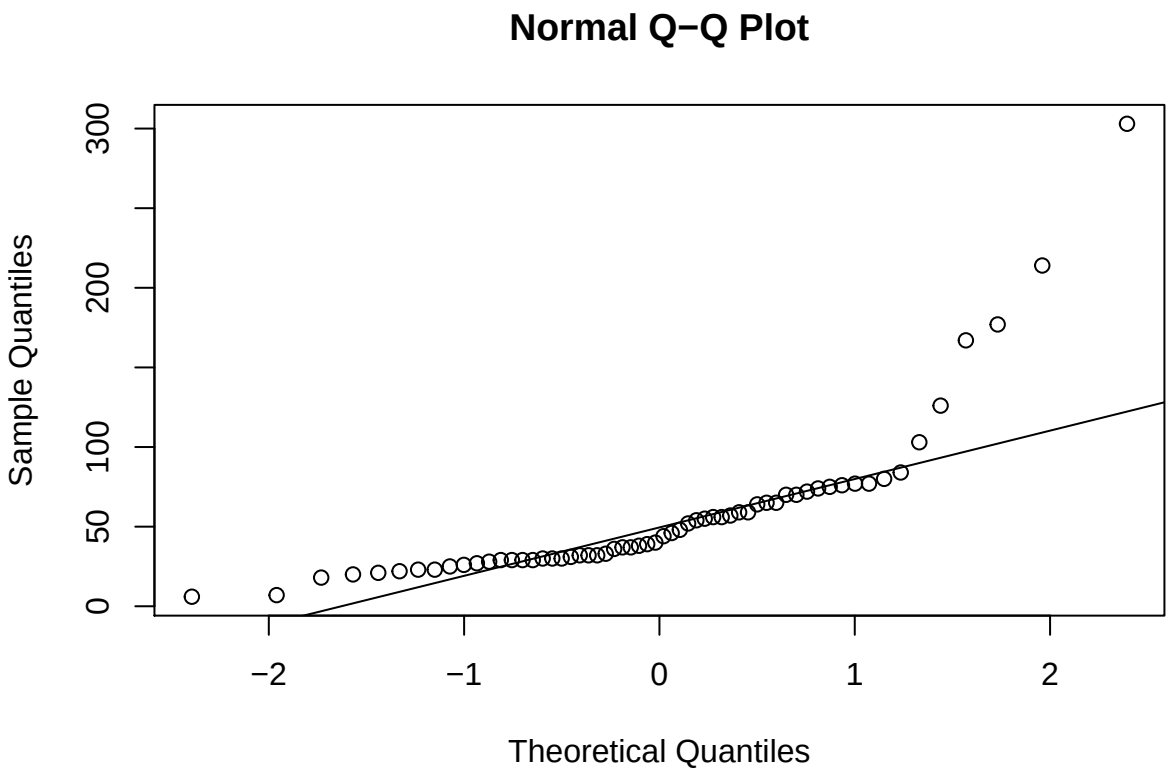
We load the data from the data source.

```
klm = scan("klm.txt");
```

```
hist(klm)
```




```
qqnorm(klm)
qqline(klm)
```



We check that the population does not seem normal. This tells us that doing the t-test is not the best option. Wilcoxon test does not seem appropiate because the population is not symmetric.

We should perform the sign test.

Task 1

We compute how many values are bigger than 31 and perform a binom test with that data.

```
n_bigger = sum(klm>31)
n = length(klm)
binom.test(n_bigger,n,p=0.5,alternative = "greater")
```

We obtain a p-value of 0.007 so we can reject the null hypothesis that the median of the population is smaller or equal than 31.

Task 2

We compute how many values are bigger than 72. We perform a binomial test with p = 0.1.

```
n_bigger = sum(klm>72);
binom.test(n_bigger,n,p=0.1,alternative = "greater")
```

With a p value of 0.006 we can reject that at most 10% of the parts arrive later than 72 days.

Exercise 4

We load the data from the data source.

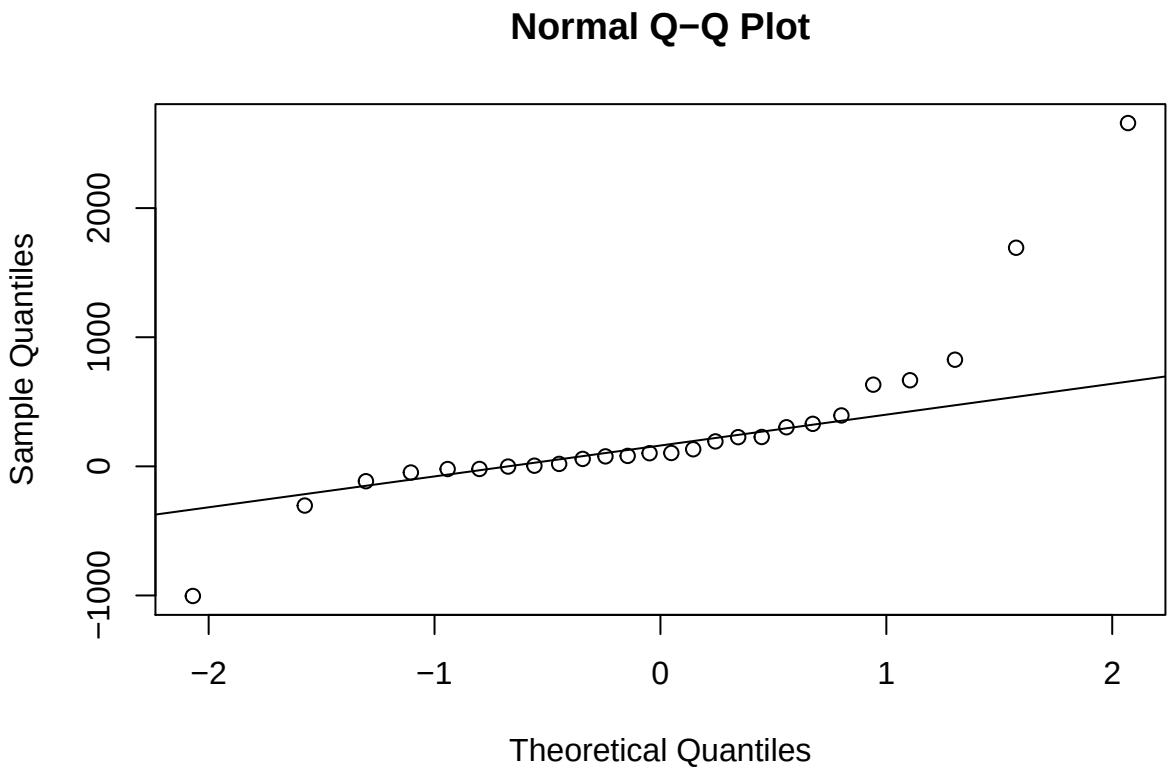
```
Clouds<-read.table("clouds.txt",header = TRUE);
```

Task 1

T-test

We check if the differences of the data come from a normal population.

```
differences=Clouds$seeded-Clouds$unseeded;
qqnorm(differences);
qqline(differences);
```



It seems that it does not come from normal population.

We perform the t-test.

```
t.test(Clouds$seeded,Clouds$unseeded,paired = TRUE)
```

The p-value 0.044 indicates that the means of both populations are different so the nitrate has an effect.

Mann-Whitney-Wilcoxon

We perform the test.

```
wilcox.test(Clouds$seeded,Clouds$unseeded)
```

The p-value 0.014 indicates that the two samples come from different populations. Therefore, the nitrate has an effect.

Kolmogorov-Smirnov

We perform the test.

```
ks.test(Clouds$seeded,Clouds$unseeded);
```

The p-value 0.019 indicates that the two samples come from different populations. Therefore, the nitrate has an effect.

Findings

The t-test seems appropiate because the data is paired but when we check the normality in the differences, it appears that the distribution is not normal so the t-test is doubtful. Since the size of the sample is small, maybe it could be the reason why the distribution does not seem normal.

Mann-Whitney and Kolmogorov-Smirnov do not seem appropiate because those test are ment to be use when the data is independent but they are not.

Task 2

We assign the square-root values of clouds to a new variable.

```
clouds_sq=sqrt(Clouds)
```

We perform the test on this variable.

```
t.test(clouds_sq$seeded,clouds_sq$unseeded,paired = TRUE)
```

The p-value 0.013 indicates that we can reject the null hypothesis.

```
wilcox.test(clouds_sq$seeded,clouds_sq$unseeded)
```

The p-value 0.014 indicates that we can reject the null hypothesis.

```
ks.test(clouds_sq$seeded,clouds_sq$unseeded)
```

The p-value0.019 indicates that we can reject the null hypothesis.

Findings :

Applying square root to the samples, when we perform t-test we can see that the p-value is being decreased. While on the other hand, in both Mann-whitney-wilcoxon test and Kolmorov-sminorv test p-value remains the same for the square rooted values.

Task 3

We assign the square-root values of the square-root of clouds to a new variable.

```
clouds_sq_sq=sqrt(clouds_sq)
```

We perform the tests on this variable.

```
t.test(clouds_sq_sq$seeded,clouds_sq_sq$unseeded,paired = TRUE)
```

The p-value 0.009 indicates that we can reject the null hypothesis, H0.

```
wilcox.test(clouds_sq_sq$seeded,clouds_sq_sq$unseeded)
```

The p-value 0.014 indicates that we can reject our null hypothesis, Ho.

```
ks.test(clouds_sq_sq$seeded,clouds_sq_sq$unseeded)
```

The p-value 0.019 indicates that we can reject our null hypothesis, Ho.

Findings :

Applying square root to the square-rooted values of the samples, when we perform t-test we can see that again the p-value is being decreased. While on the other hand, in both Mann-Whitney-Wilcoxon test and Kolmorov-Sminorv test p-value remains the same for the square rooted values.

Exercise 5

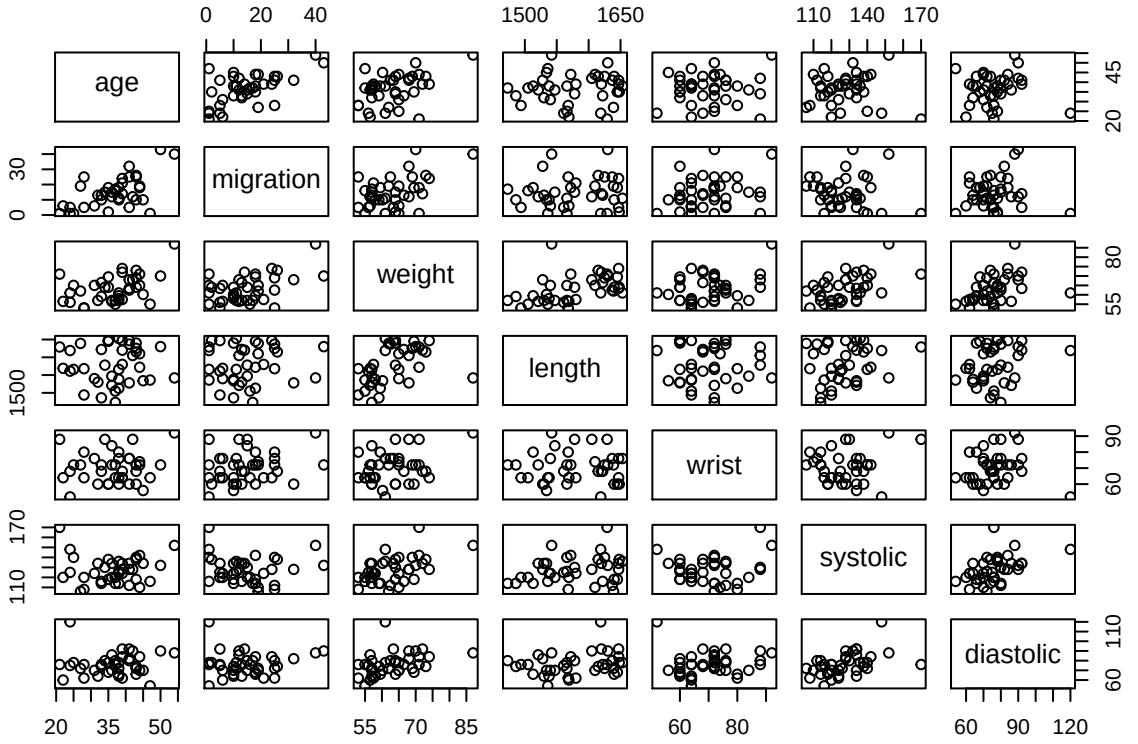
We load the data from the data source.

```
peruvians = read.table("peruvians.txt", header = TRUE);
peruvians = peruvians[, - c(5, 6, 7)] # Deleting redundant columns since we don't need to use
attach(peruvians)
```

Task 1

We make the plot of each pair of two variables.

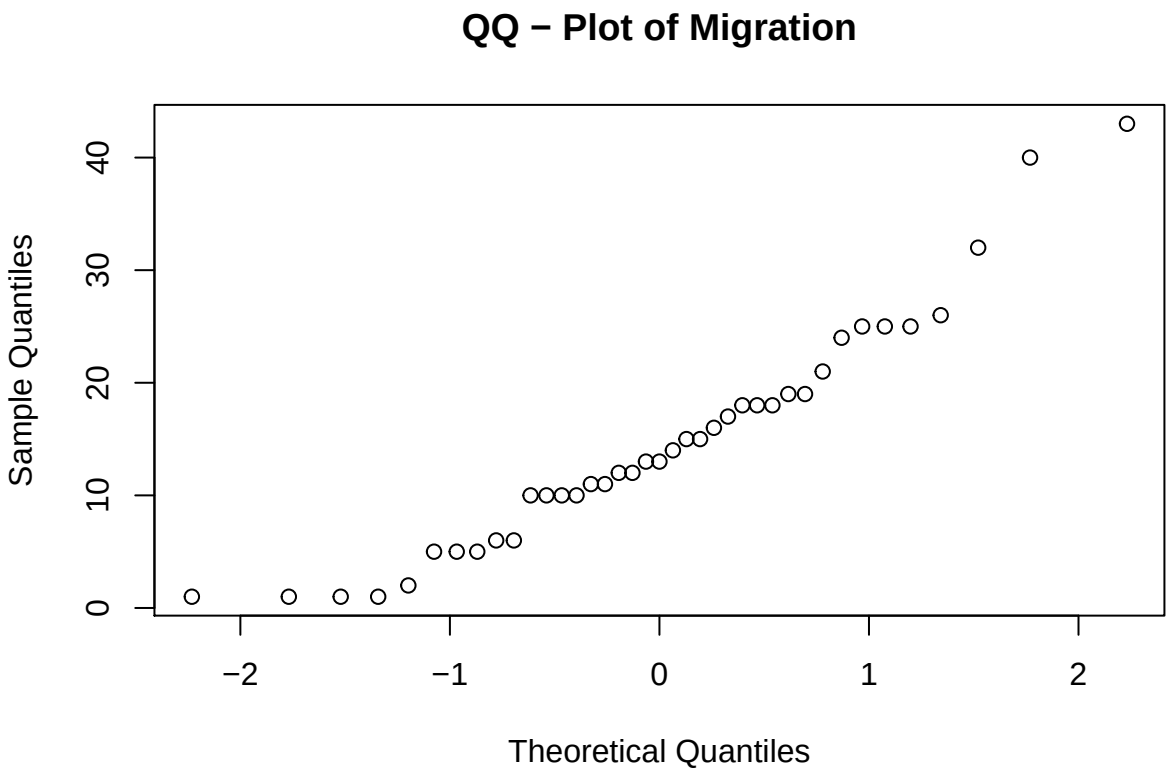
```
pairs(peruvians)
```



From this plot we can see that migration is probably correlated with age and weight because when the values of migration increase, then the values of age and weight also increase.

Task 2

First, we check the normality of variable migration. After finding the normality, then we can decide which test should we perform for correlation checking.




```
shapiro.test(migration)
```

The shapiro test gives a p-value of Shapiro-Wilk normality test which rejects the null hypothesis that the distribution is normal. From the above test result and from QQ-Plot, we found that migration does not belong to normal distribution. Therefore, we will use the rank correlation test of Spearman.

```
cor.test(migration, age, method = "spearman")
```

The p-value 0.002 indicates that the null hypothesis that the correlation is 0 should be rejected. Therefore, migration and age are correlated.

```
cor.test(migration, weight, method = "spearman")
```

The p-value 0.029 indicates that the null hypothesis that the correlation is 0 should be rejected. Therefore, migration and weight are correlated.

```
cor.test(migration, length, method = "spearman")
```

The p-value 0.609 indicates that we cannot reject that the null hypothesis that the correlation is 0. Therefore, migration and length are not correlated.

```
cor.test(migration, wrist, method = "spearman")
```

The p-value 0.18 indicates that we cannot reject that the null hypothesis that the correlation is 0. Therefore, migration and heart rate are not correlated.

```r
cor.test(migration, systolic, method = "spearman")
```

The p-value 0.305 indicates that we cannot reject that the null hypothesis that the correlation is 0. Therefore, migration and systolic are not correlated.

```r
cor.test(migration, diastolic, method = "spearman")
```

The p-value 0.649 indicates that we cannot reject that the null hypothesis that the correlation is 0. Therefore, migration and diastolic are not correlated.

Exercise 6

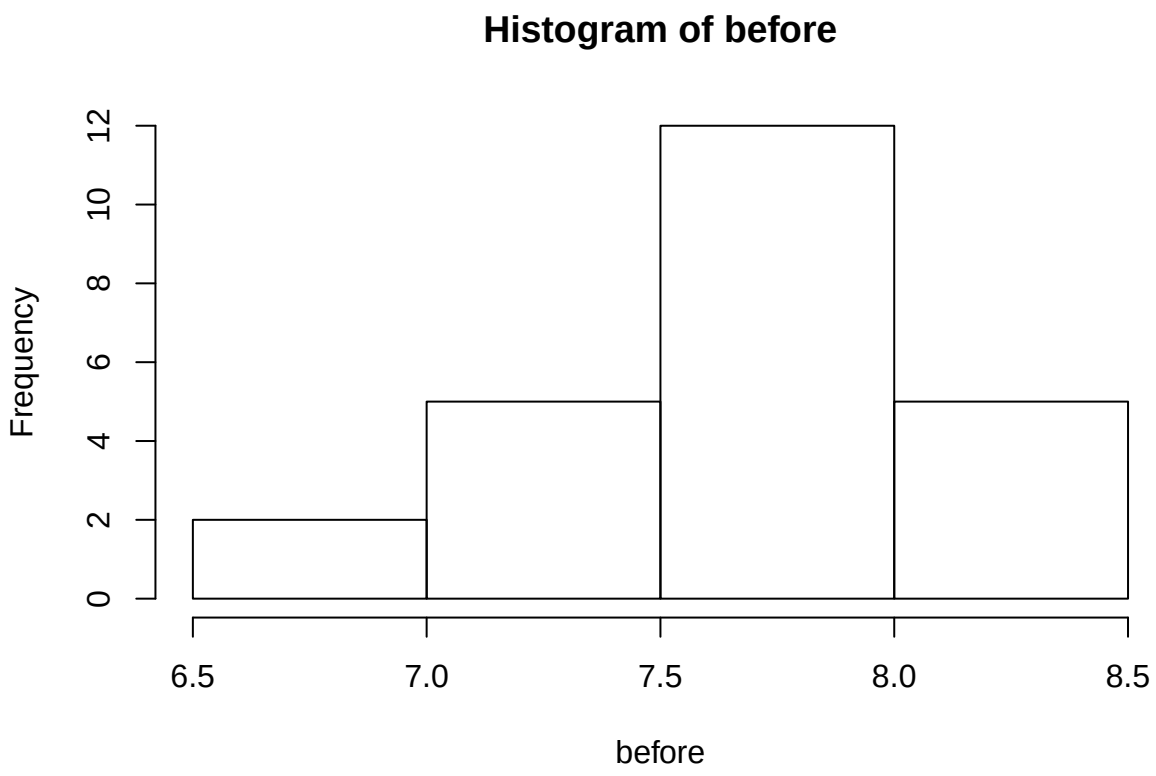
First we get the data from the data source

```r
run = read.table("run.txt", header = TRUE)
attach(run)
```

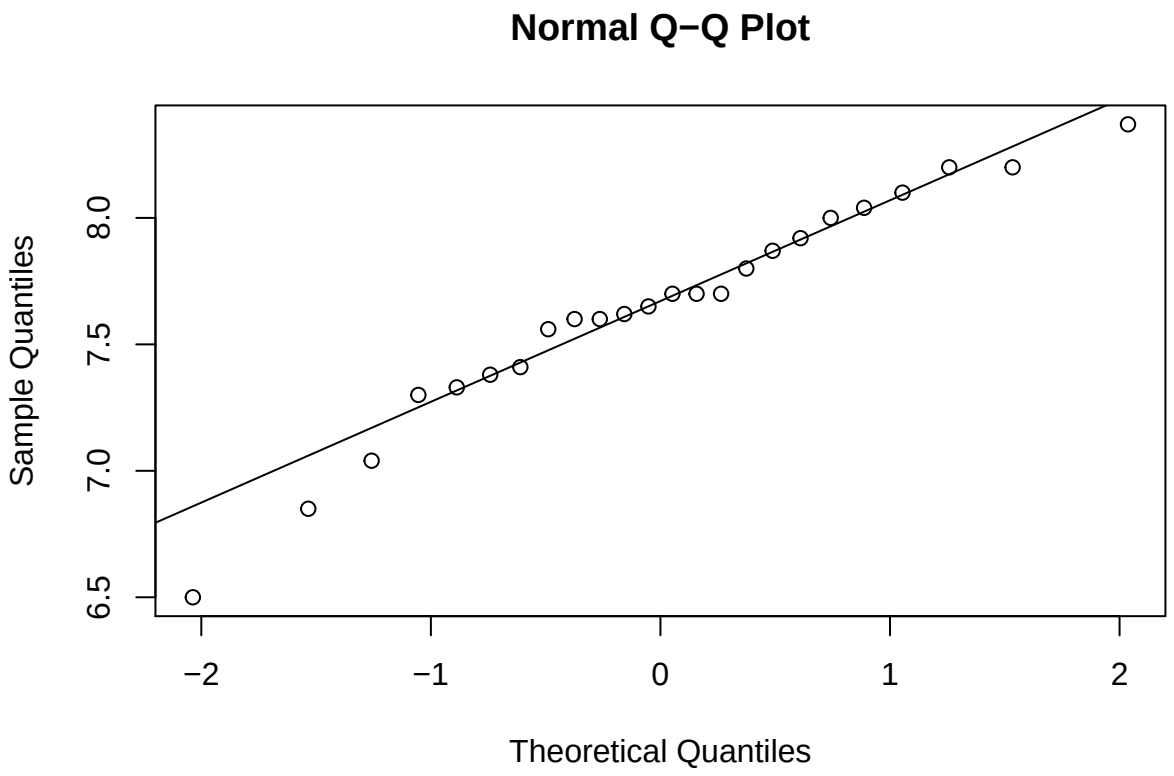
Task 1

Here we show histogram and QQ-plot of the data before taking the drinks.

```r
hist(before)
```



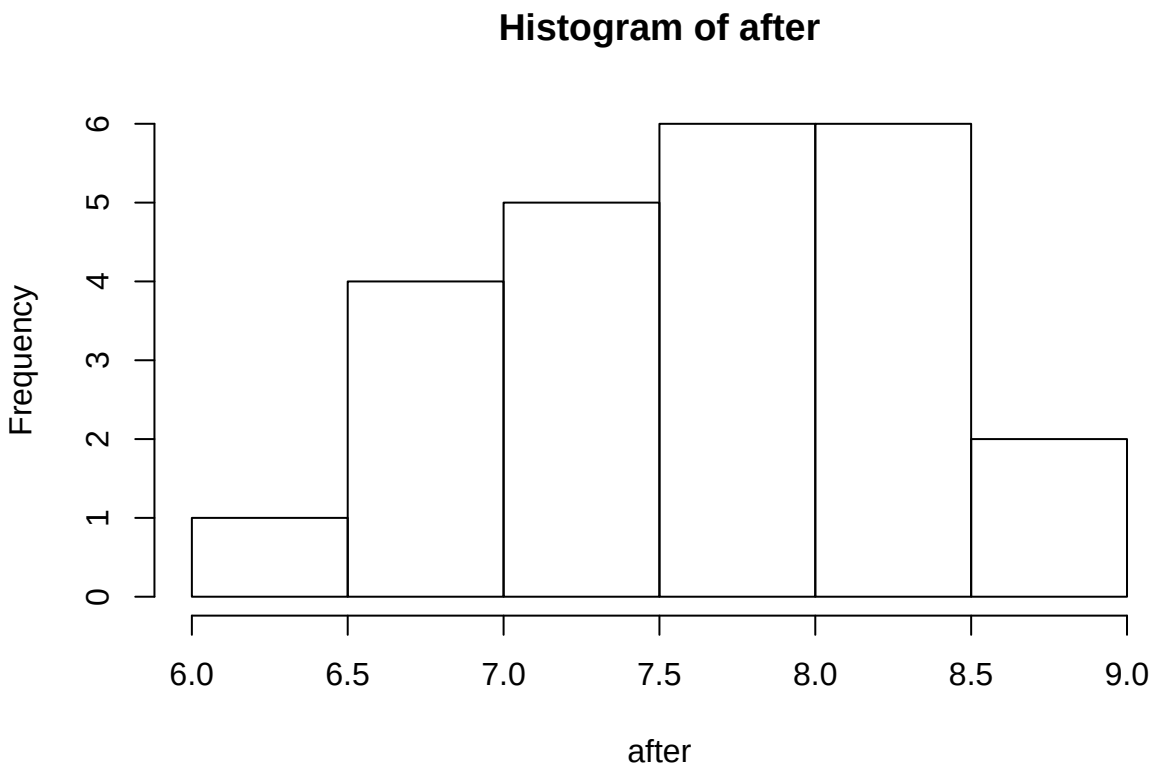
```
qqnorm(before)
qqline(before)
```



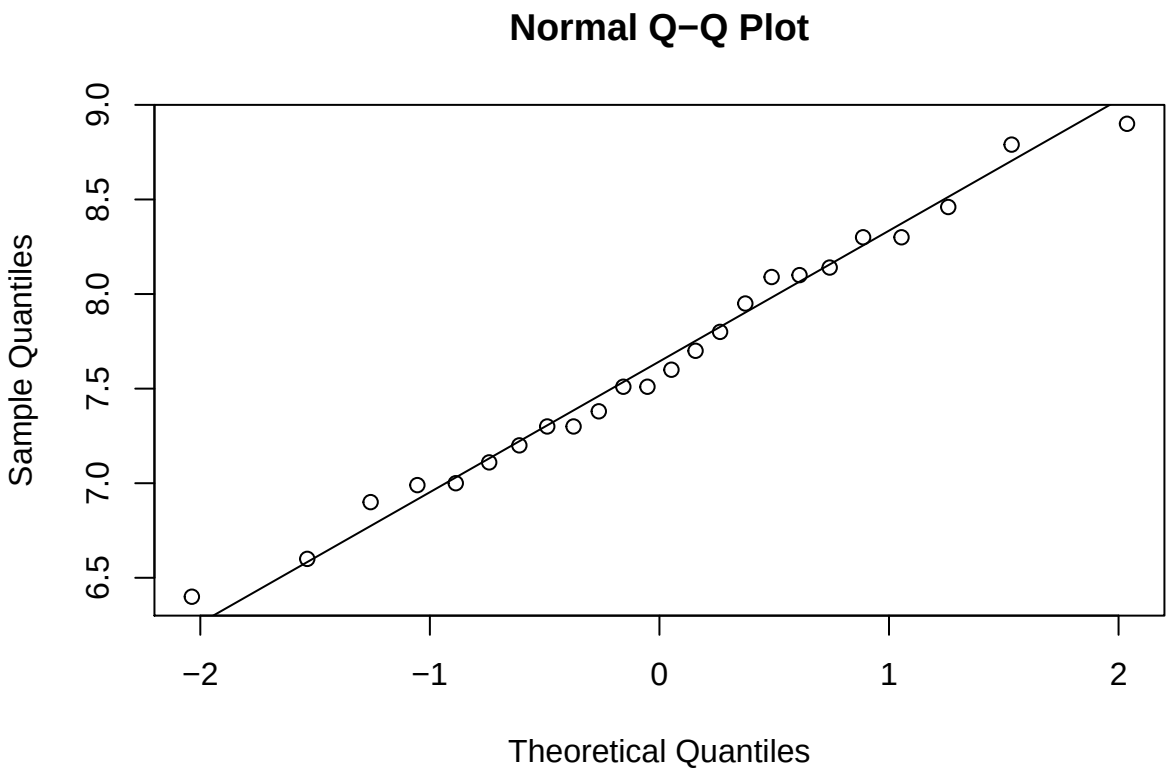
The data seems a bit deviated from the normal distribution.

We generate the same kind of plots of the data after taking the drinks.

```
hist(after)
```



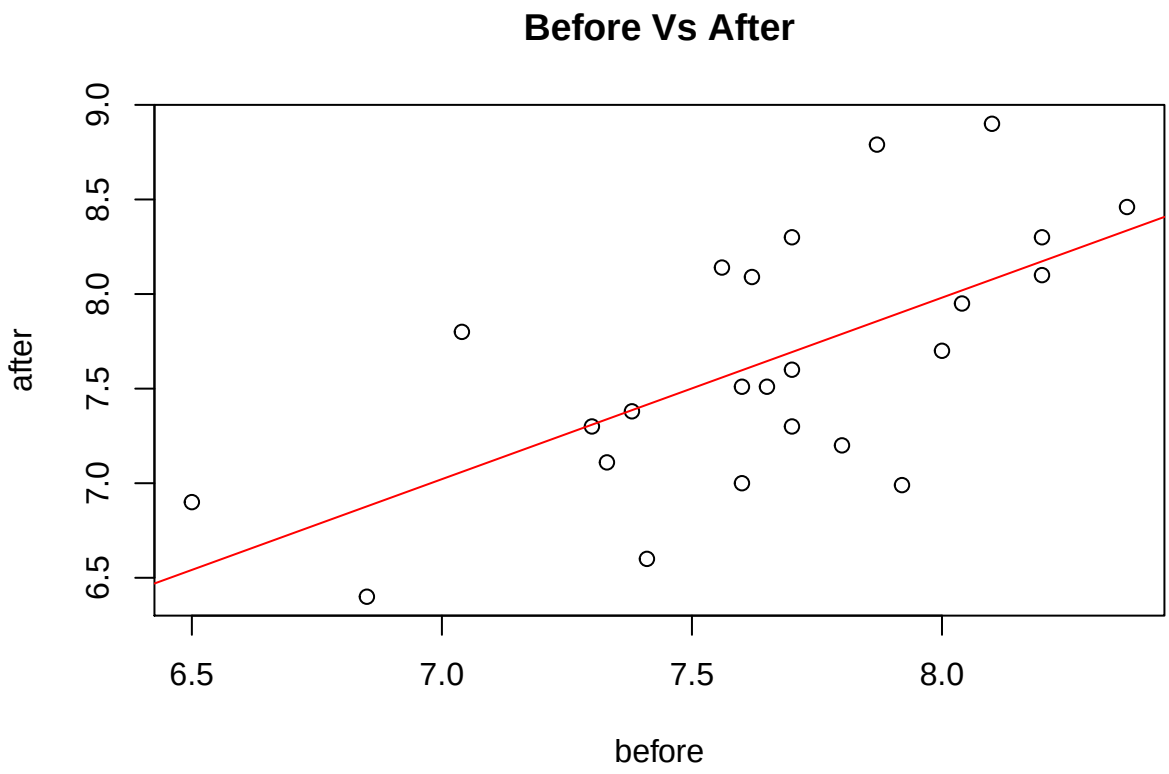

```
qqnorm(after)
qqline(after)
```



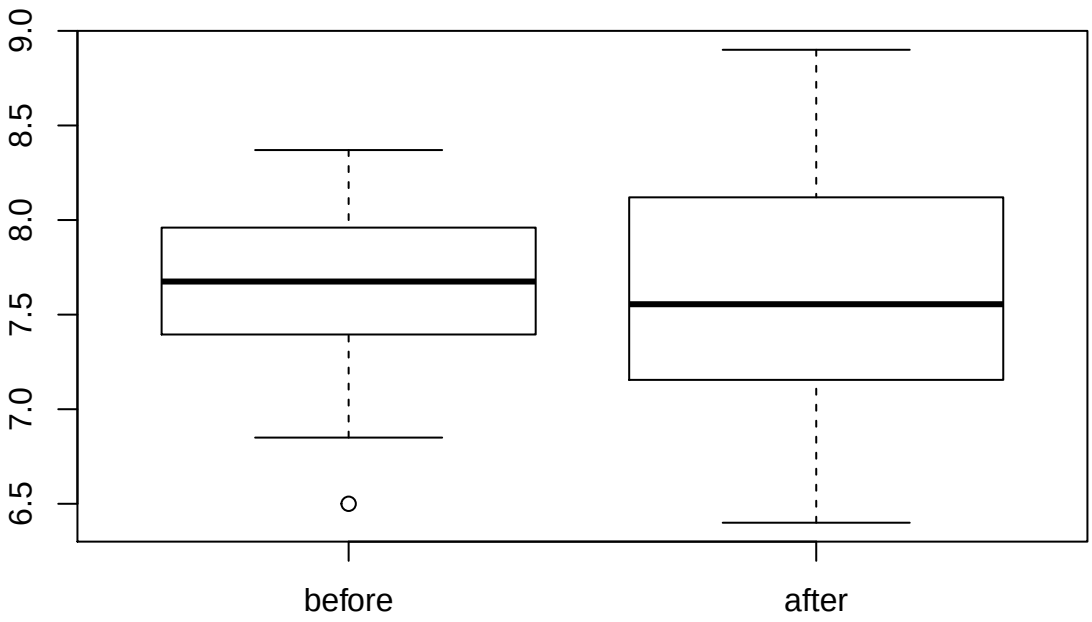
The QQ-plot shows data the data follows the normal distribution.

Then we plot both data together in order to be able to see if there is correlation.

```
plot(before, after, main = "Before Vs After")
abline(lm(after~before), col="red")
```



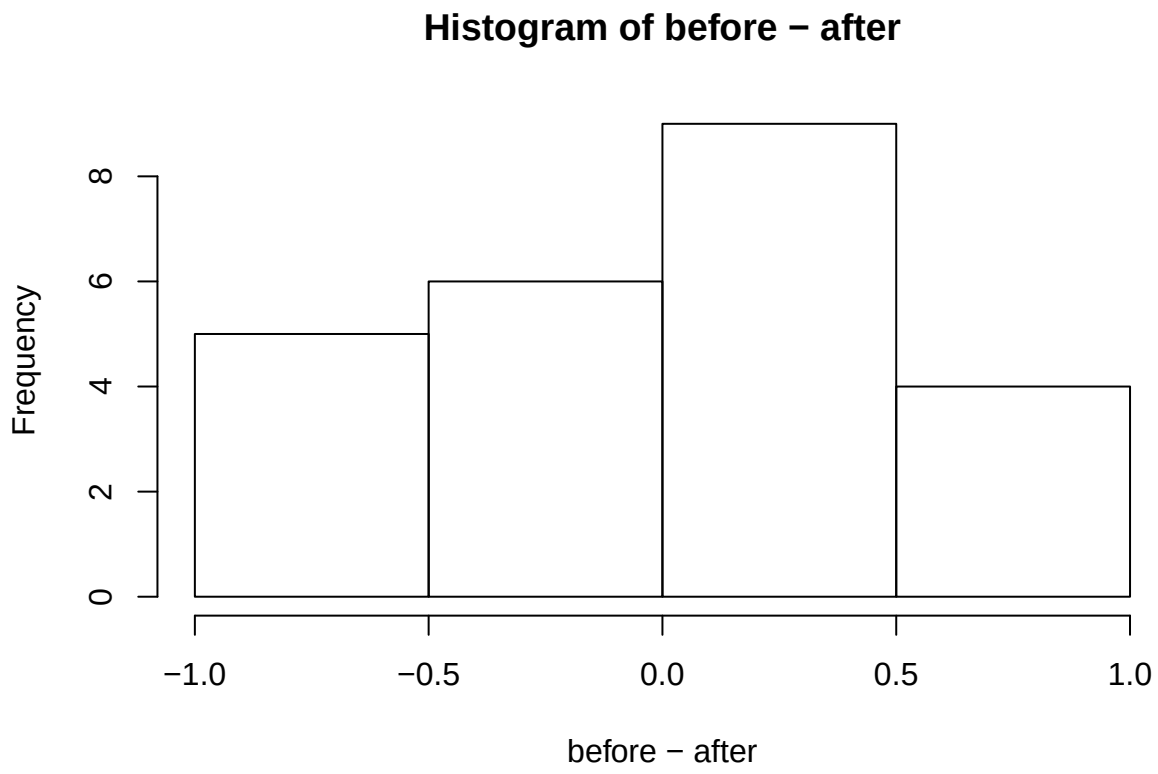

```
boxplot(before, after, names = c("before", "after"))
```



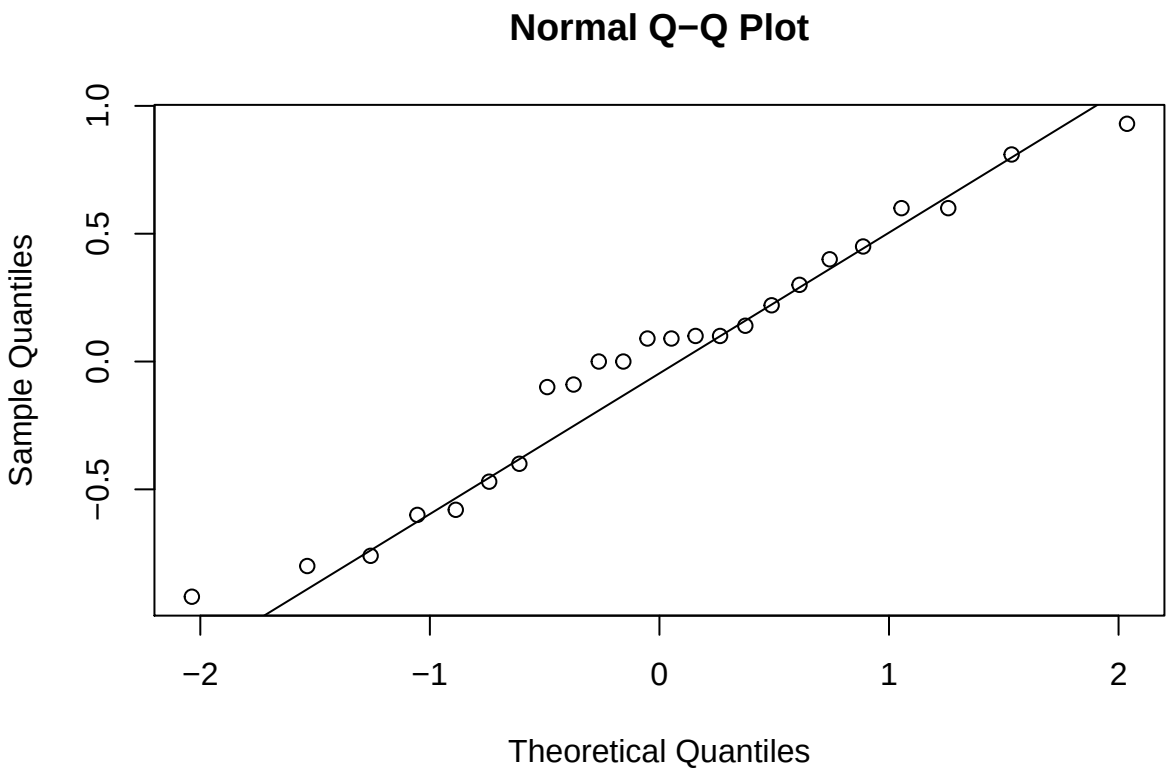
It seems that there is correlation between the two data sets.

Finally we plot the differences of the two datasets in a histogram and QQ-plot.

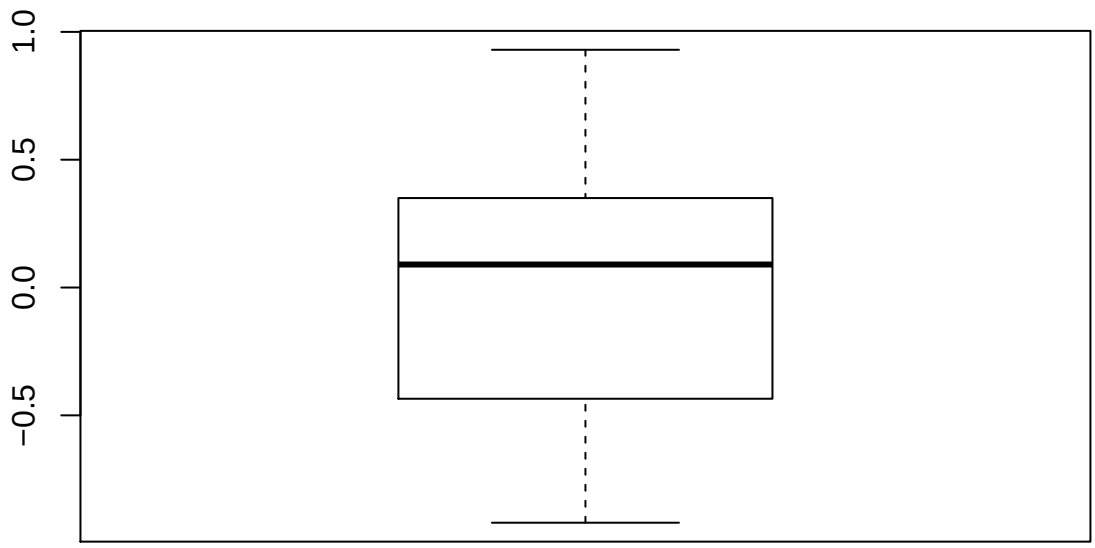
```
hist(before-after)
```




```
qqnorm(before-after)
qqline(before-after)
```




```r
boxplot(before-after, names = "before - after")
```



The histogram and QQ-plot show that the data probably comes form the normal distribution.

Task 2

Since t-test runs under the assumption that the differences of the samples come from a normal distribution, we check the normality using the shapiro test.

```
shapiro.test(before-after)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  before - after
## W = 1, p-value = 0.6
```

The test proves the assumption that the difference is normal.

We assign the different times to new variables for the soft drink and energy drink.

```
lemo = run[which(drink == "lemo"),]
energy = run[which(drink == "energy"),]
```

Now, we test the data before having the soft drink with the one after.

```r
t.test(lemo$before, lemo$after, paired = TRUE)
```

Having the p-value 0.437 we cannot reject that the mean of the differences is 0. Therefore, we can conclude the soft drinks do not affect the runners.

Now we do the same test for the energy drinks.

```r
t.test(energy$before, energy$after, paired = TRUE)
```

Having the p-value 0.126 we cannot reject that the mean of the differences is 0. Therefore, we can conclude the energy drinks do not affect the runners.

Task 3

Now we test individually if the time differences are affected by the drink type.

```r
lemo_time_difference = lemo$before - lemo$after

energy_time_difference = energy$before - energy$after

t.test(lemo_time_difference, energy_time_difference)
```

Having the p-value 0.159 we come to the conclusion that we cannot reject that the mean of the differences is 0. Furthermore, we can not say that the time difference is affected by the type of drink.

Task 4

A possible objection could be that the time taken between the two runs are not enough. According to http://www.sciencefocus.com/article/human-body/how-long-does-caffeine-take-kick, caffeine takes around 45 minutes to kick in. So, maybe half an hour was not enough to see the effects of the drinks in the running speed.

Task 5

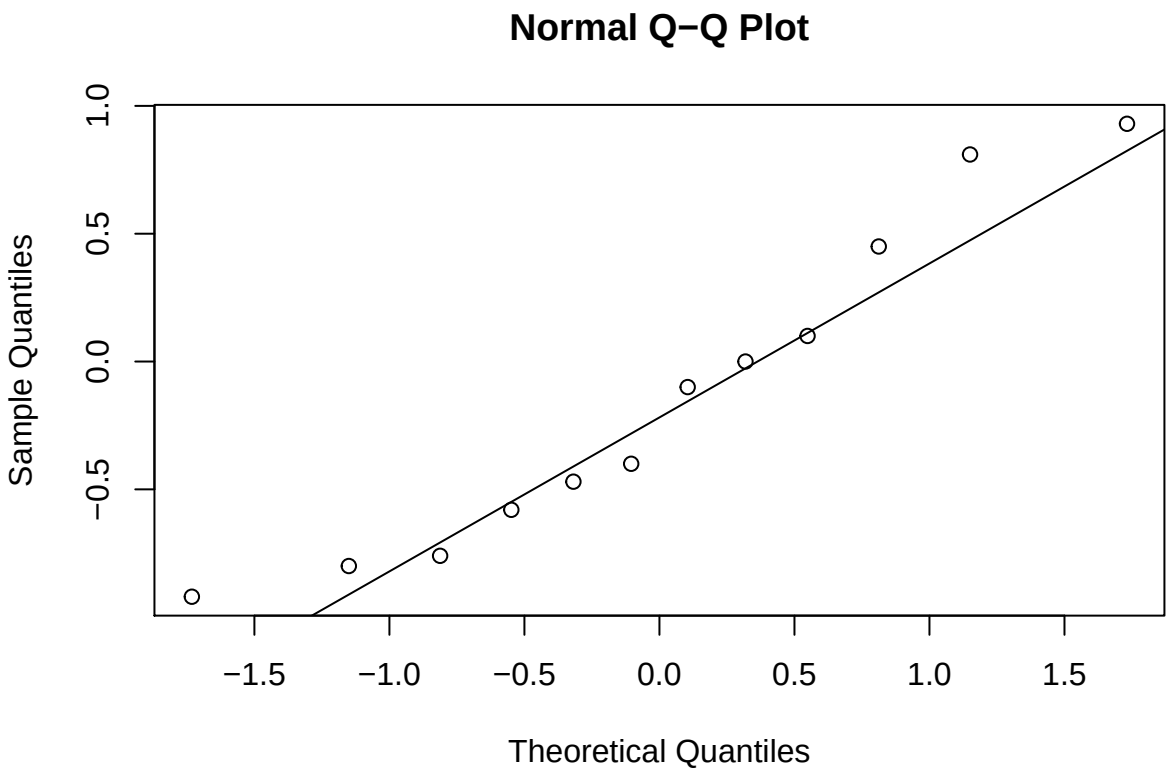
Our possible objection could be that 30 minutes is not enough time to rest and do the second run in the same conditions as the first run.

Task 6

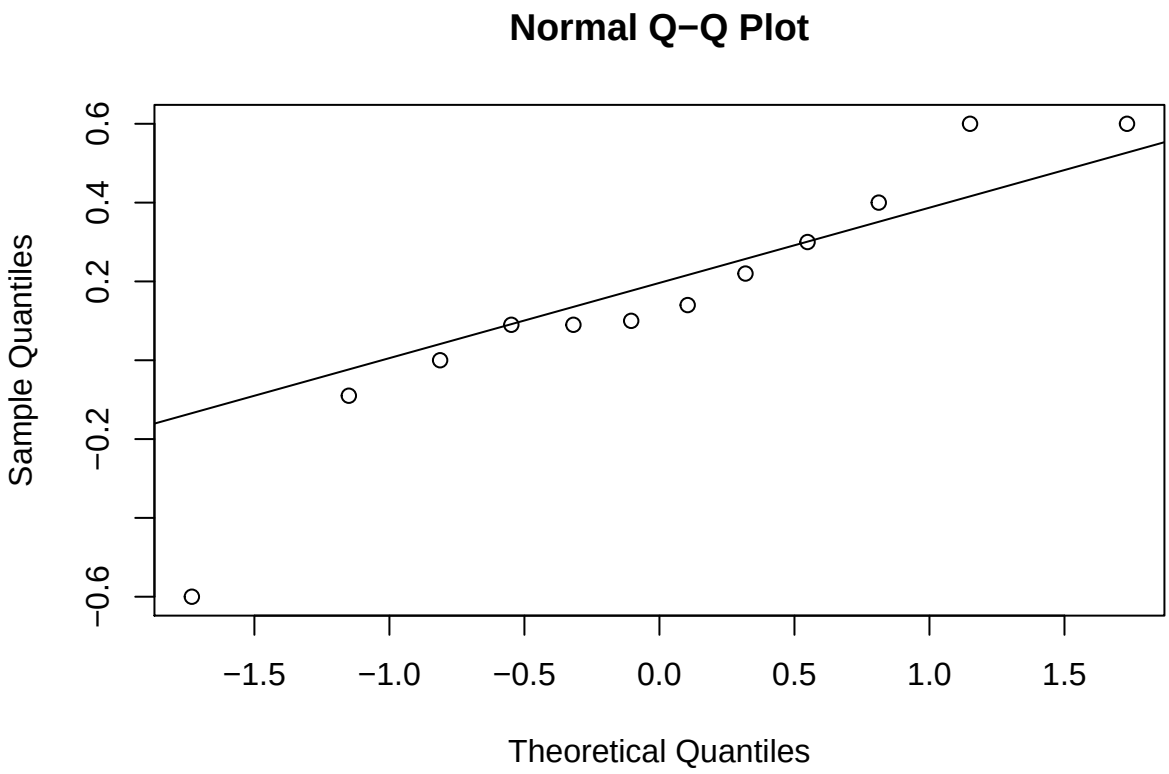
The test assumes that the distribution of the differences is normal.

Here we show the QQ-plots for the time difference for the soft drink and the energy drink.

```r
qqnorm(lemo_time_difference)
qqline(lemo_time_difference)
```




```
qqnorm(energy_time_difference)
qqline(energy_time_difference)
```



Exercise 7

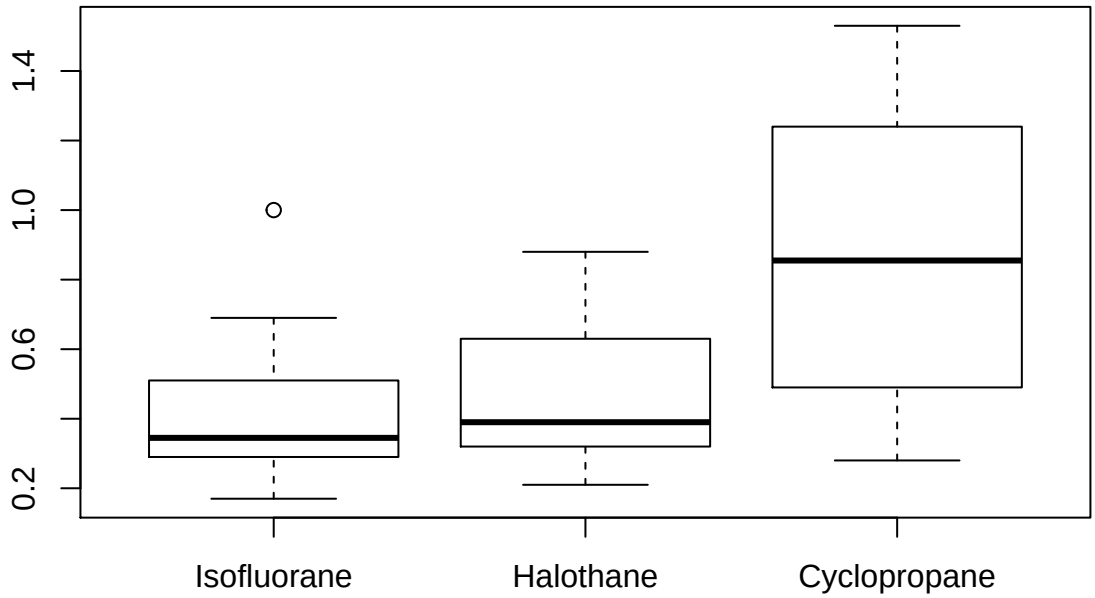
First, we load the data from the data source.

```
dogs <- read.table("dogs.txt", header = TRUE)
attach(dogs)
```

Task 1

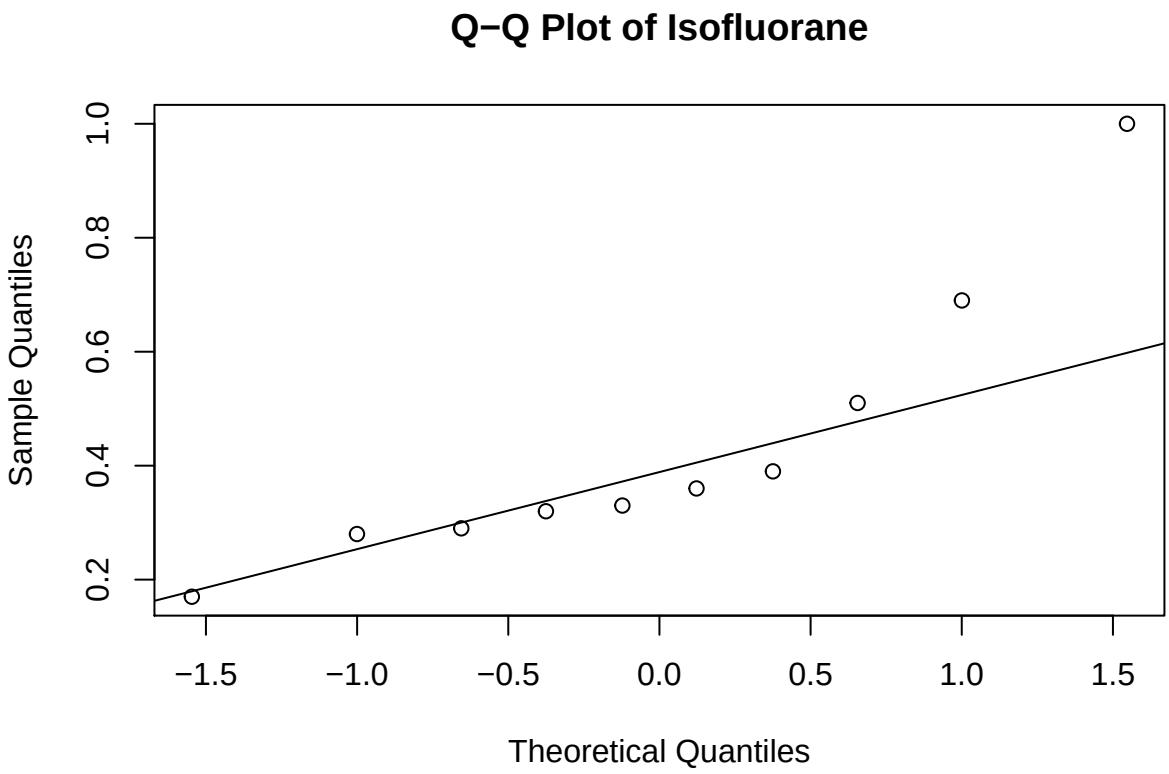
First of all, we start with boxplots of all the different drugs on the population.

```
boxplot(dogs, names = c("Isofluorane", "Halothane", "Cyclopropane") )
```



Then we check the normality on the distributions of each drug. Showing the QQ-plots first and then performing the shapiro test.

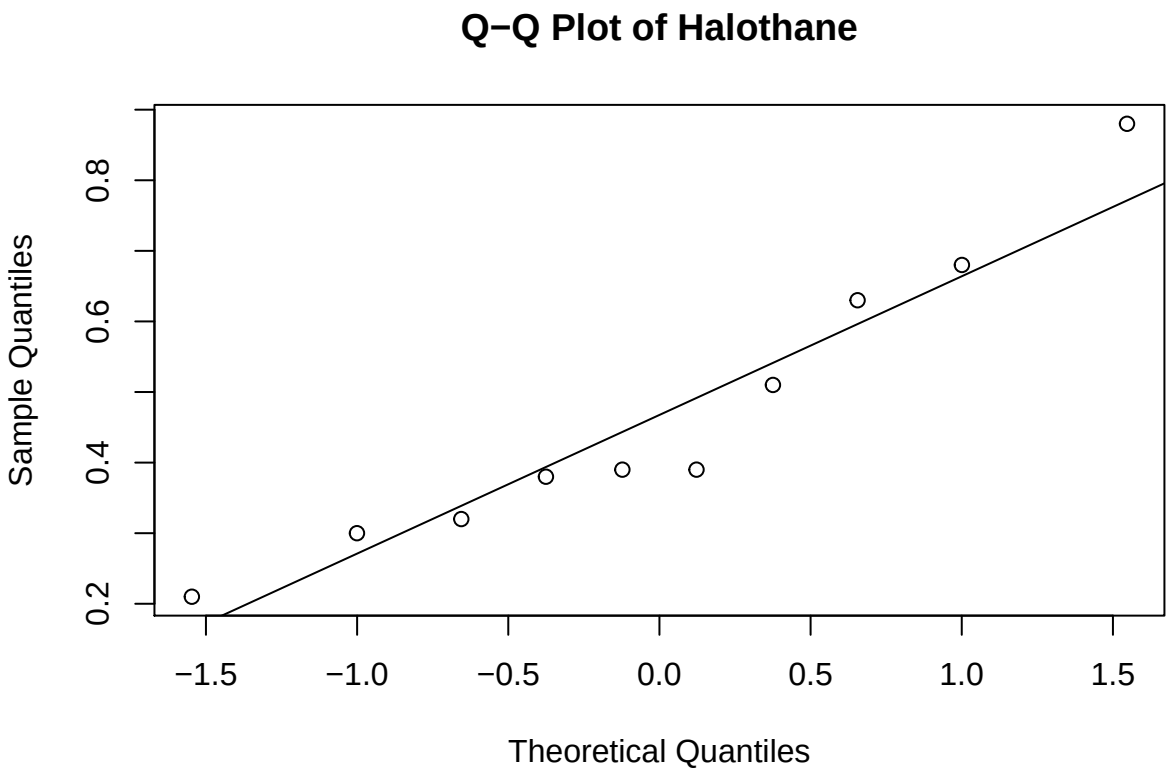
```
qqnorm(isofluorane, main= "Q-Q Plot of Isofluorane" )
qqline(isofluorane)
```




```
shapiro.test(isofluorane)
```

It it not reasonable to assume that the Isofluorane data was taken from a normal population, having the p-value 0.034.

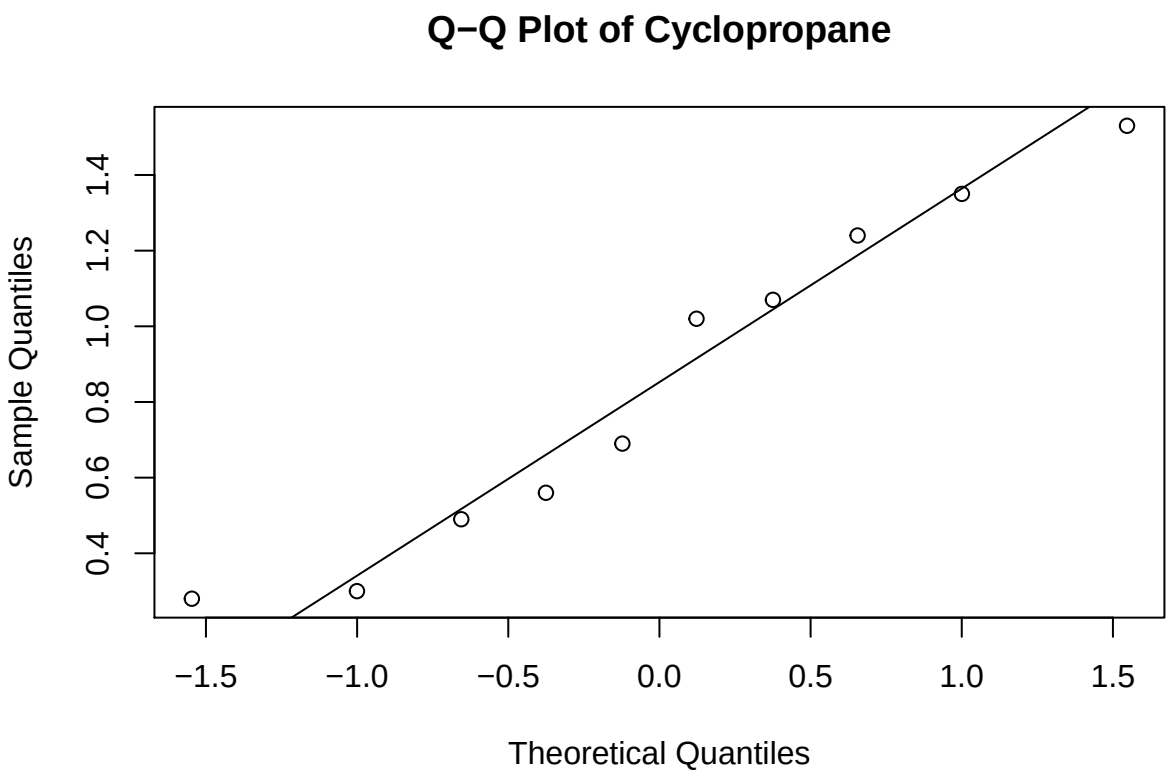
```
qqnorm(halothane, main= "Q-Q Plot of Halothane" )
qqline(halothane)
```




```
shapiro.test(halothane)
```

Taking a look at the p-value 0.386 we can say it is reasonable to assume that the population of Halothane was taken from a normal population.

```
qqnorm(cyclopropane, main= "Q-Q Plot of Cyclopropane" )
qqline(cyclopropane)
```




```
shapiro.test(cyclopropane)
```

Taking a look at the p-value 0.482 we can say it is reasonable to assume that the population of Halothane was taken from a normal population.

Task 2

```
dogframe = data.frame(plasma=as.vector(as.matrix(dogs)),  group = factor(rep(1:3, each=10)))
dog_aov = lm(plasma~group, data = dogframe)
anova(dog_aov)

summary(dog_aov)
```

Having the p-value from the anova test being 0.011, NA we can reject that mu0 = mu1 = mu2. Therefore, the concentration is not the same under the different drugs.

running the summary(dog_aov) test, we get that the estimate for Isofluorane is 0.434 the estimate for Halothane is 0.469 and the estimate for Cyclopropane is 0.853.

Task 3

```
attach(dogframe)
kruskal.test(plasma, group)
```

The test gives us the p-value 0.059. Which indicates that we cannot reject that m0 = m1 = m2.

So we can say the results of this test differ from the conclussion made after the anova test.