Assignment 1

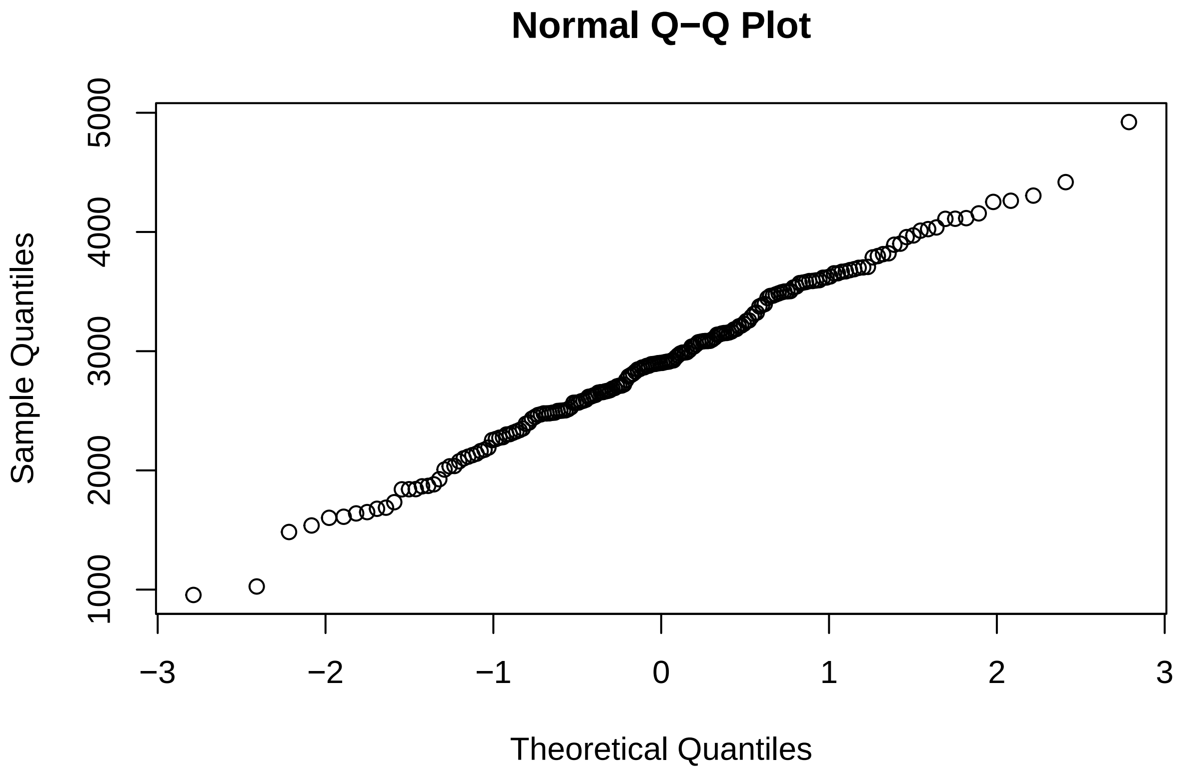
Andrei Udriste, Xinyu Hu, Maria Gheorghe-Tudor - Group 43

2021/2/19

Exercise 1. *Birthweights*

The data set birthweight.txt contains the birthweights of 188 newborn babies. We are interested in finding the underlying (population) mean mu of birthweights. a) Check normality of the data. Compute a point estimate for mu. Derive, assuming normality (irrespective of your conclusion about normality of the data), a bounded 90% confidence interval for mu.

```
data = read.table(file="birthweight.txt", header=TRUE)#Read the data from the file
qqnorm(data$birthweight)#Create a QQ plot to check the normality of the data
```



As it can be observed, the QQ-plot creates a straight line, this means that the data has a normal distribution.

```
data_mean = mean(data$birthweight)#Compute the estimate point for mu
cat("mu =", data_mean)
```

```
## mu = 2913.293
```

It can be observed that the value of mu is equal with 2913.293.

```
error = qt(0.95, df=length(data$birthweight) - 1) * sd(data$birthweight) / sqrt(length(data$birthweight
upper_bound = data_mean + error#Compute the upper bound of the confidence interval
```

```r
lower_bound = data_mean - error#Compute the lower bound of te confidence interval
cat("90% confidence interval = (", lower_bound,",", upper_bound, ")")
```

```
## 90% confidence interval = ( 2829.202 , 2997.384 )
```

After computing the confidence interval, it can be observed that 90% of the data is between 2829.202 and 2997.384.

b) An expert claims that the mean birthweight is bigger than 2800, verify this claim by using a t-test. What is the outcome of the test if you take alpha = 0.1? And other values of alpha?

```r
t.test(data$birthweight, mu=2800, alternative="greater")#Compute the t-test to check if the mean is big
```

```
##
##  One Sample t-test
##
## data:  data$birthweight
## t = 2.2271, df = 187, p-value = 0.01357
## alternative hypothesis: true mean is greater than 2800
## 95 percent confidence interval:
##  2829.202      Inf
## sample estimates:
## mean of x
##  2913.293
```

If an alpha of 0.1 is chosen then hypothesis H0 is rejected and H1 is accepted. In this case the mean of the data is bigger than 2800. The same result can be obtained if alpha is bigger than 0.013. If alpha has a value that is lower than 0.013 then hypothesis H0 will be accepted and H1 will be rejected.

c) In the R-output of the test from b), also a confidence interval is given, but why is it different from the confidence interval found in a) and why is it one-sided?

```r
t.test(data$birthweight)
```

```
##
##  One Sample t-test
##
## data:  data$birthweight
## t = 57.269, df = 187, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  2812.939 3013.646
## sample estimates:
## mean of x
##  2913.293
```

```r
t.test(data$birthweight, conf.level=0.9)
```

```
##
##  One Sample t-test
##
## data:  data$birthweight
## t = 57.269, df = 187, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  2829.202 2997.384
## sample estimates:
## mean of x
## 2913.293
```

Exercise 2. *Power function of the t-test*

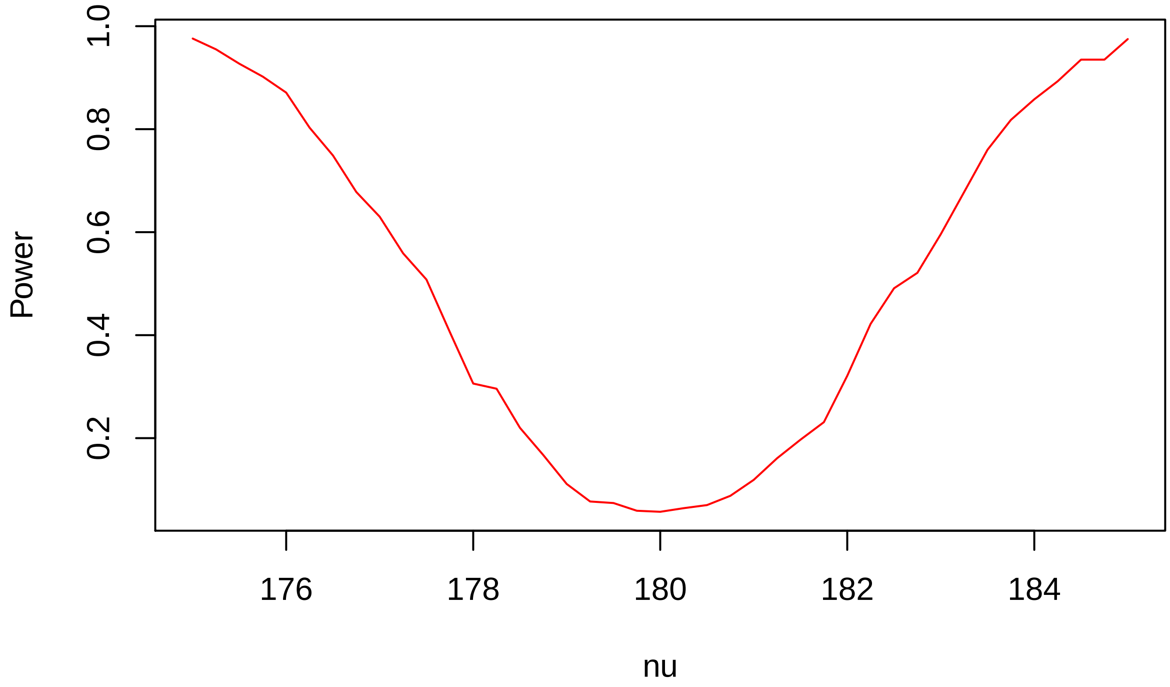
We study the power function of the two-sample t-test (see Section 1.9 of Assignment 0). For n=m=30, mu=180, nu=175 and sd=5, generate 1000 samples x=rnorm(n,mu,sd) and y=rnorm(m,nu,sd), and record the 1000 p-values for testing H 0 : mu=nu. You can evaluate the power (at point nu=175) of this t-test as fraction of p-values that are smaller than 0.05.

```
#Initialize the variables that are going to be used to calculate the power
n = m = 30
mu = 180
nu = 175
sd = 5
B = 1000
p = numeric(B)

#Calculate the p-value for a "B" number of times
for(b in 1:B){
  x = rnorm(n, mu, sd)#Initialize "n" data points from a normal distribution with a mean "mu" and a sta
  y = rnorm(m, nu, sd)#Initialize "m" data points from a normal distribution with a mean "nu" and a sta
  p[b] = t.test(x, y, var.equal=TRUE)[[3]]#Obtain the p-value after doing the t-test
}
power = mean(p<0.05)#Calculate the power function
cat("The power of the test is equal with:", power)
```

```
## The power of the test is equal with: 0.969
```

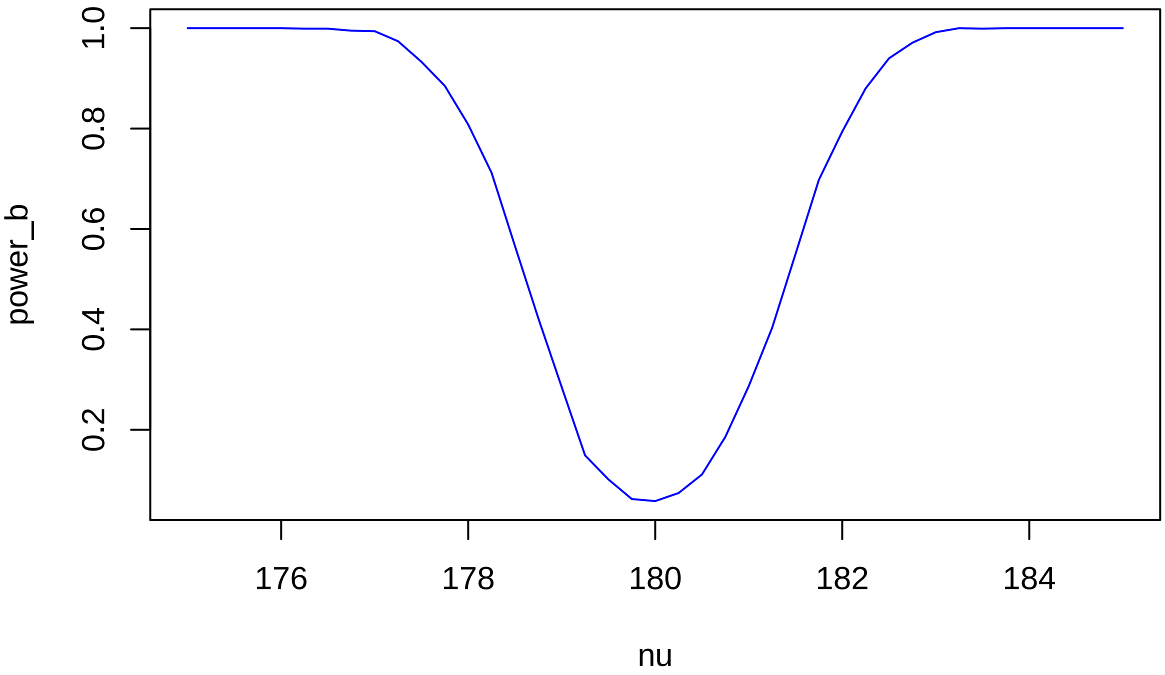
a) Set n=m=30, mu=180 and sd=5. Calculate now the power of the t-test for every value of nu in the grid seq(175,185,by=0.25). Plot the power as a function of nu.

```
#Initialize the variables that are going to be used to calculate the power
n = m = 30
mu = 180
nu = 175
sd = 5
B = 1000
p = numeric(B)
nu = seq(175, 185, by=0.25)
power_a = numeric(length(nu))

#Calculate the power for each "nu" value
for(i in 1:length(nu)){
  #Calculate the p-value for a "B" number of times
  for(b in 1:B){
    x = rnorm(n, mu, sd)#Initialize "n" data points from a normal distribution with a mean "mu" and a s
    y = rnorm(m, nu[i], sd)#Initialize "m" data points from a normal distribution with a mean "nu" and
    p[b] = t.test(x, y, var.equal=TRUE)[[3]]#Obtain the p-value after doing the t-test
  }
  power_a[i] = mean(p<0.05)#Calculate the power function of a particular value of "nu"
}
plot(nu, power_a, type="l", col="red", ylab="Power")#Plot the power as a function of "nu"
```



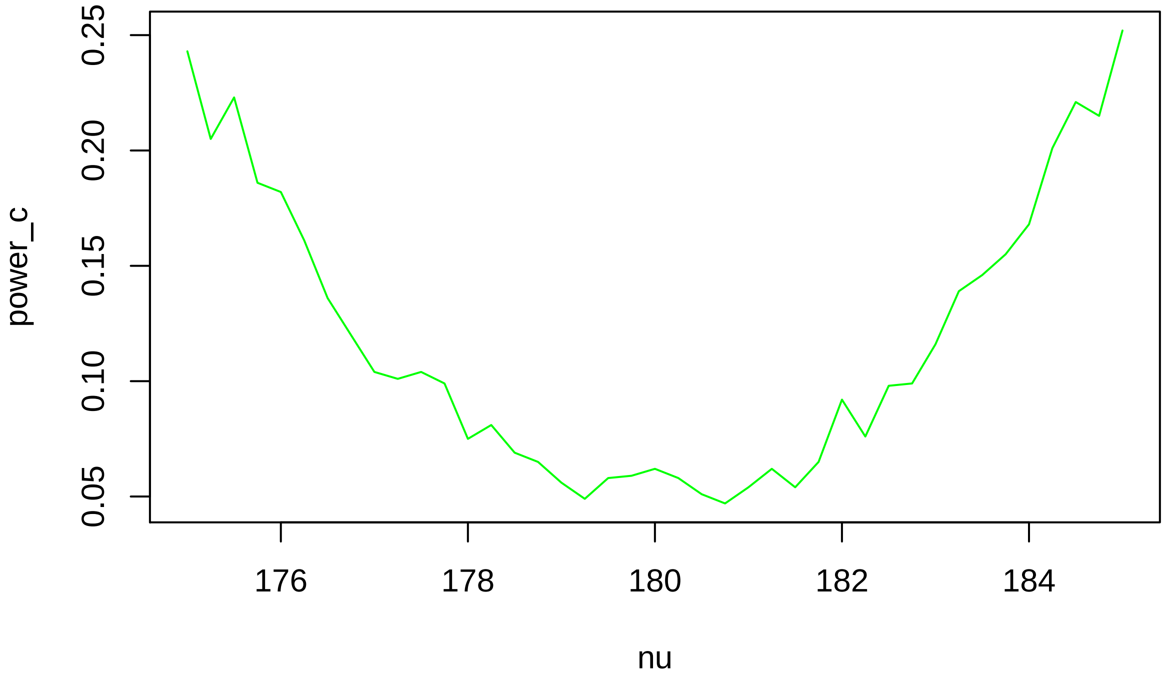
b) Set n=m=100, mu=180 and sd=5. Repeat the preceding exercise. Add the plot to the preceding plot.

```r
#Initialize the variables that are going to be used to calculate the power
n = m = 100
mu = 180
nu = 175
sd = 5
B = 1000
p = numeric(B)
nu = seq(175, 185, by=0.25)
power_b = numeric(length(nu))

#Calculate the power for each "nu" value
for(i in 1:length(nu)){
  #Calculate the p-value for a "B" number of times
  for(b in 1:B){
    x = rnorm(n, mu, sd)#Initialize "n" data points from a normal distribution with a mean "mu" and a s
    y = rnorm(m, nu[i], sd)#Initialize "m" data points from a normal distribution with a mean "nu" and
    p[b] = t.test(x, y, var.equal=TRUE)[[3]]#Obtain the p-value after doing the t-test
  }
  power_b[i] = mean(p<0.05)#Calculate the power function of a particular value of "nu"
}
plot(nu, power_b, type="l", col="blue")#Plot the power as a function of "nu"
```



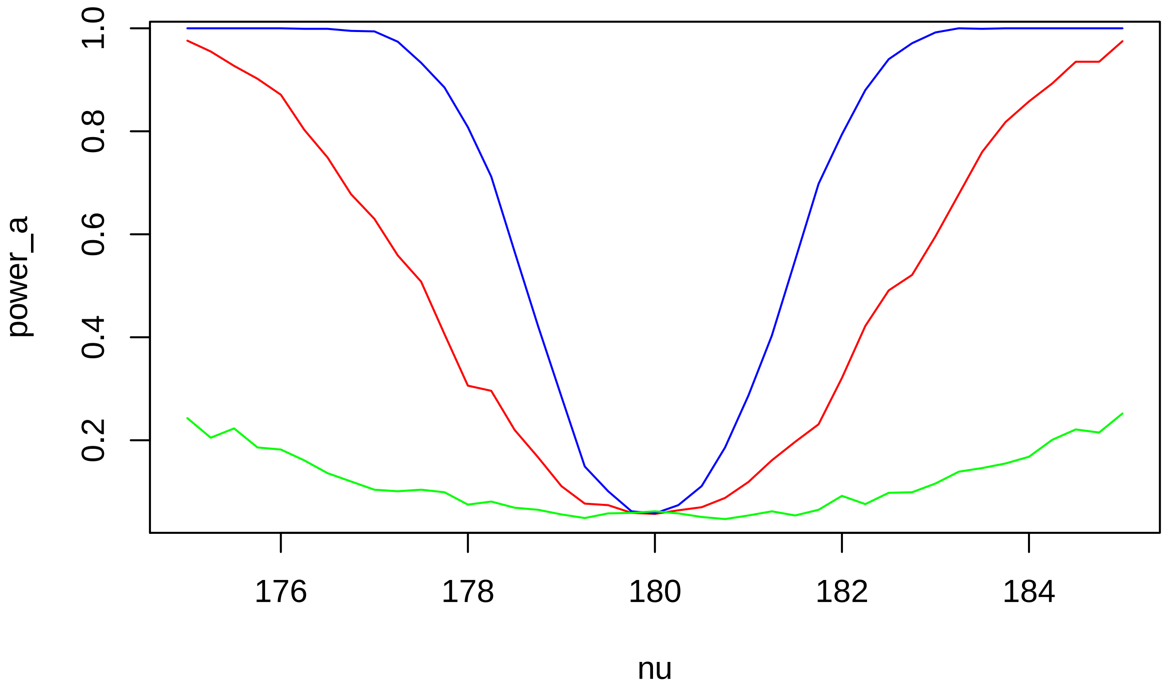
c) Set n=m=30, mu=180 and sd=15. Repeat the preceding exercise.

```r
#Initialize the variables that are going to be used to calculate the power
n = m = 30
mu = 180
nu = 175
sd = 15
B = 1000
p = numeric(B)
nu = seq(175, 185, by=0.25)
power_c = numeric(length(nu))

#Calculate the power for each "nu" value
for(i in 1:length(nu)){
  #Calculate the p-value for a "B" number of times
  for(b in 1:B){
    x = rnorm(n, mu, sd)#Initialize "n" data points from a normal distribution with a mean "mu" and a s
    y = rnorm(m, nu[i], sd)#Initialize "m" data points from a normal distribution with a mean "nu" and
    p[b] = t.test(x, y, var.equal=TRUE)[[3]]#Obtain the p-value after doing the t-test
  }
  power_c[i] = mean(p<0.05)#Calculate the power function of a particular value of "nu"
}
plot(nu, power_c, type="l", col="green")#Plot the power as a function of "nu"
```



d) Explain your findings.

```
plot(nu, power_a, type="l", col="red")#Plot the power obtained from point "a" as a function of "nu"
points(nu, power_b, type="l", col="blue")#Plot the power obtained from point "b" as a function of "nu"
points(nu, power_c, type="l", col="green")#Plot the power obtained from point "c" as a function of "nu"
```

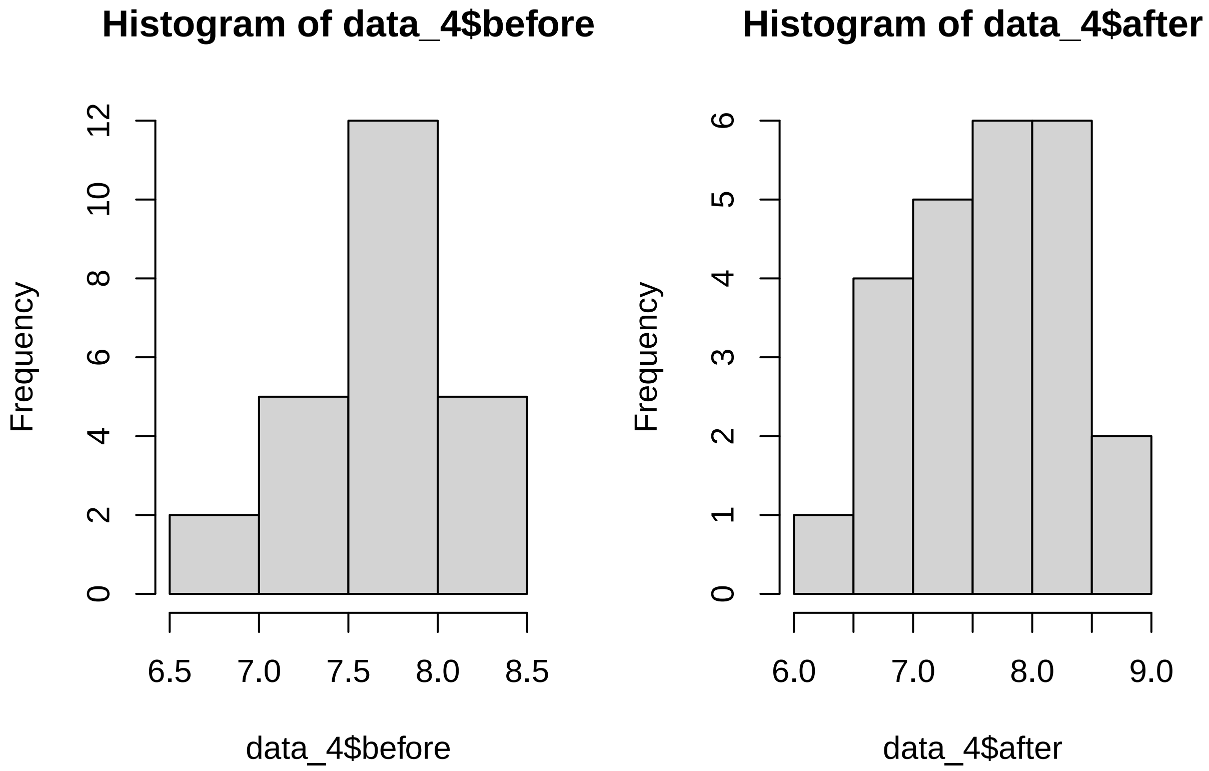


Exercise 4. *Energy drink*

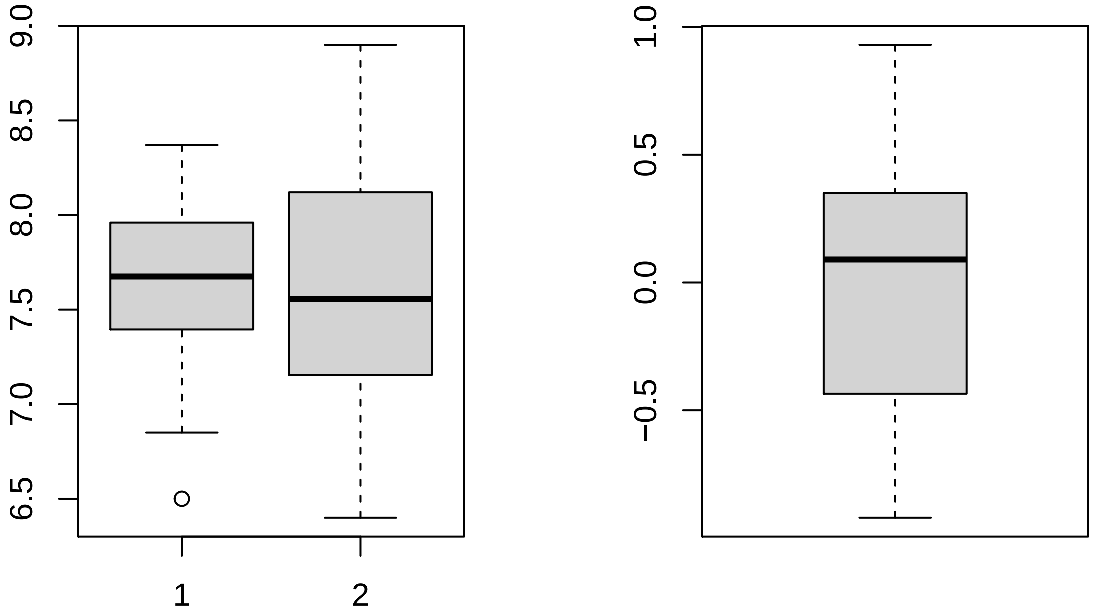
To study the effect of energy drink a sample of 24 high school pupils were randomized to drinking either a softdrink or an energy drink after running for 60 meters. After half an hour they were asked to run again. For both sprints they were asked to sprint as fast they could, and the sprinting time was measured. The data is given in the file run.txt. [Courtesy class 5E, Stedelijk Gymnasium Leiden, 2010.]

a) Disregarding the type of drink, test whether the run times before drink and after are correlated.

```
data_4 = read.table(file="run.txt", header=TRUE)
par(mfrow=c(1,2))
hist(data_4$before)
hist(data_4$after)
```



```
par(mfrow=c(1,2))
boxplot(data_4$before, data_4$after)
boxplot(data_4$before - data_4$after)
```



```
t.test(data_4$before, data_4$after, paired=TRUE)
```

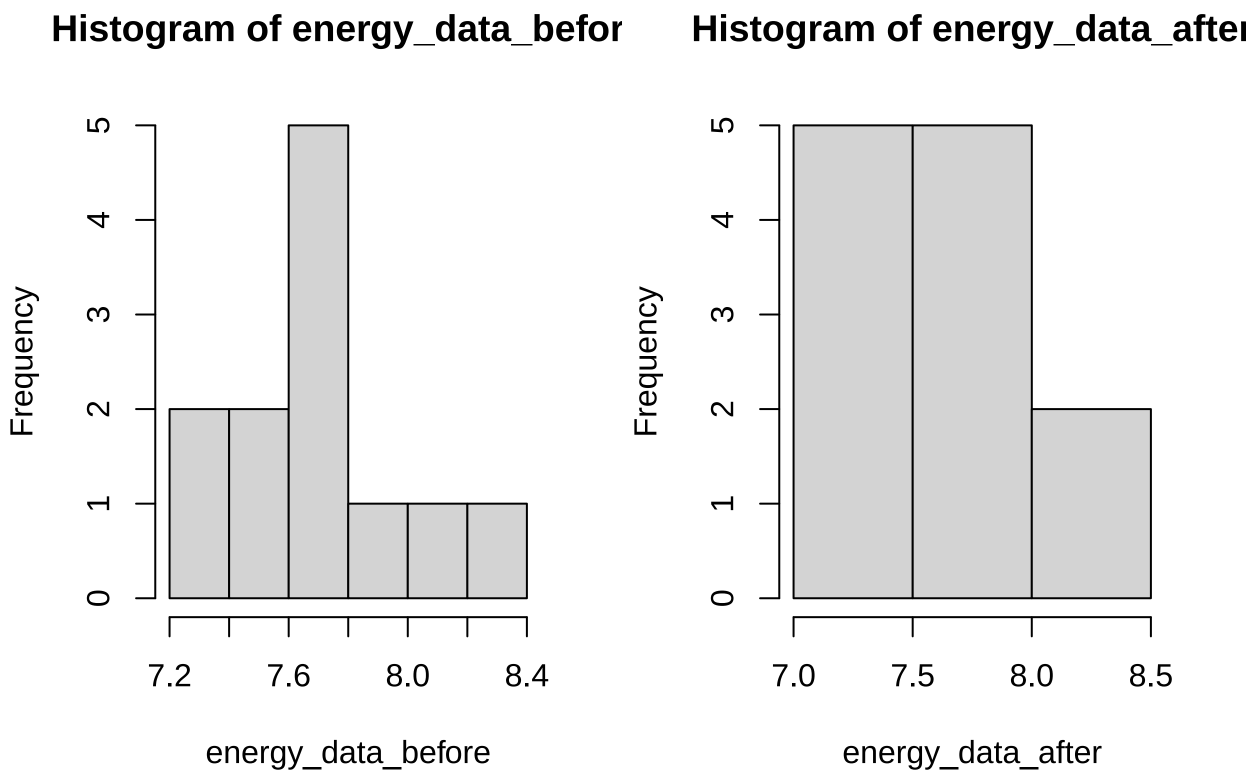
```
##
##  Paired t-test
##
## data:  data_4$before and data_4$after
## t = 0.044122, df = 23, p-value = 0.9652
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2103076  0.2194743
## sample estimates:
## mean of the differences
##             0.004583333
```

```
t.test(data_4$before - data_4$after)
```

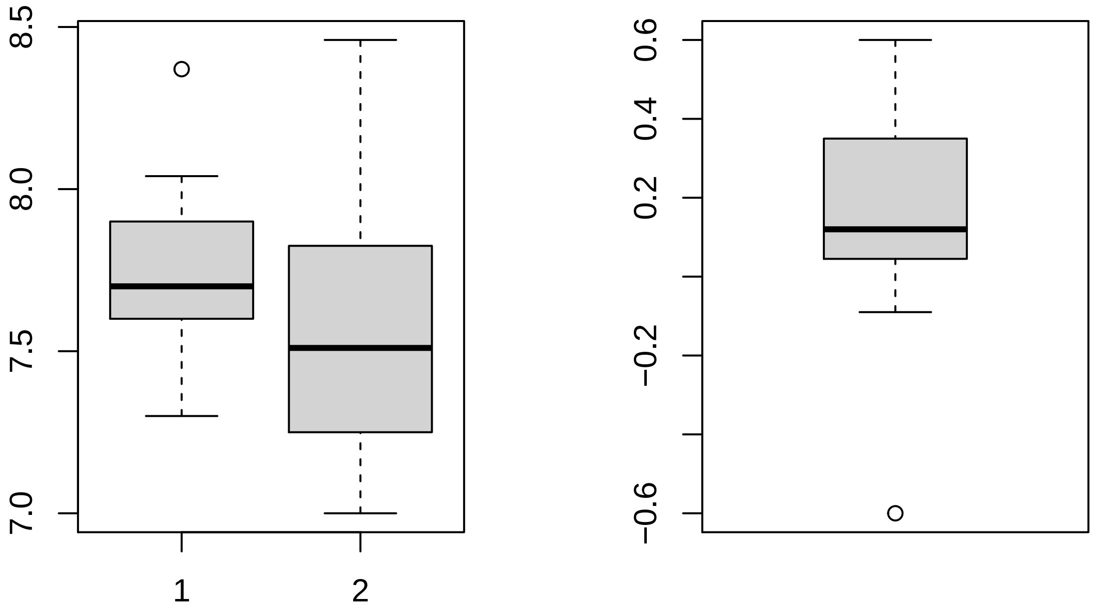
```
##
##  One Sample t-test
##
## data:  data_4$before - data_4$after
## t = 0.044122, df = 23, p-value = 0.9652
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.2103076  0.2194743
## sample estimates:
##   mean of x
## 0.004583333
```

b) Test separately, for both the softdrink and the energy drink conditions, whether there is a difference in speed in the two running tasks.

```
energy_data_before = data_4$before[data_4[,3] == "energy"]
energy_data_after = data_4$after[data_4[,3] == "energy"]
par(mfrow=c(1,2))
hist(energy_data_before)
hist(energy_data_after)
```



```
par(mfrow=c(1,2))
boxplot(energy_data_before, energy_data_after)
boxplot(energy_data_before - energy_data_after)
```



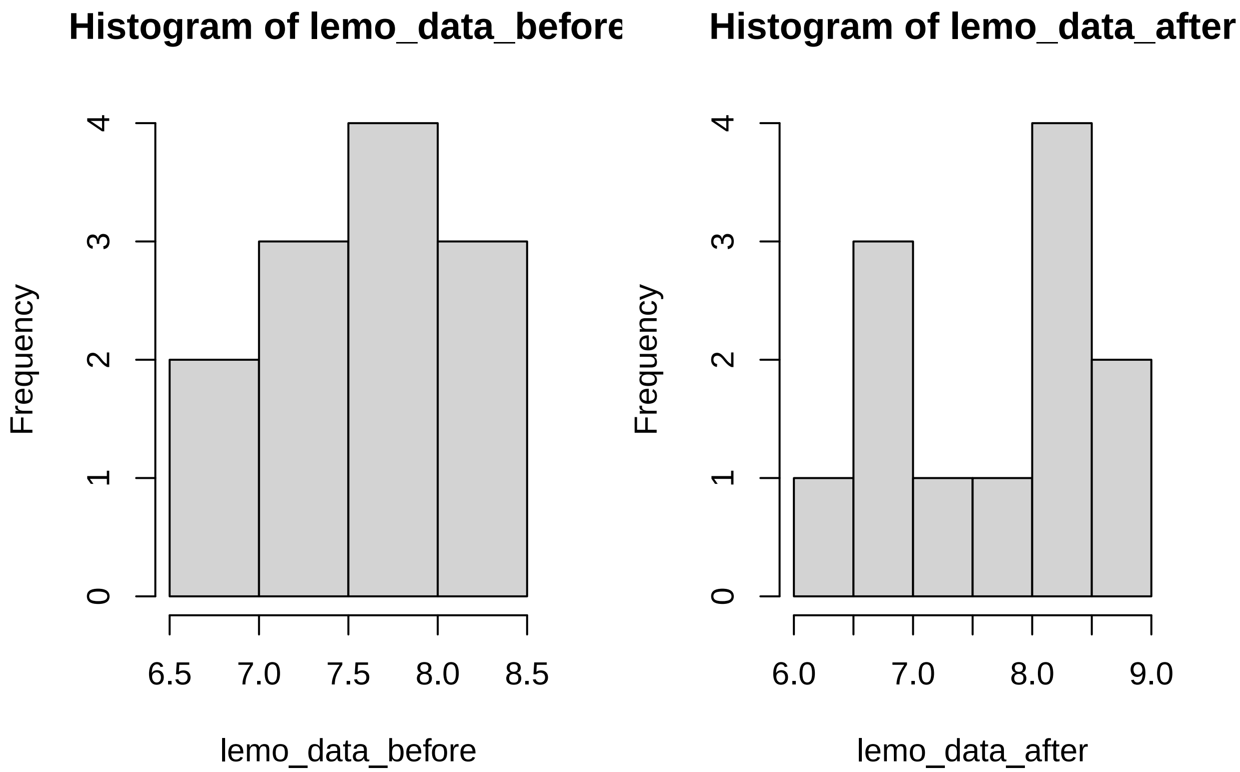
```
t.test(energy_data_before, energy_data_after, paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  energy_data_before and energy_data_after
## t = 1.6538, df = 11, p-value = 0.1264
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.05101059  0.35934392
## sample estimates:
## mean of the differences
##               0.1541667
```

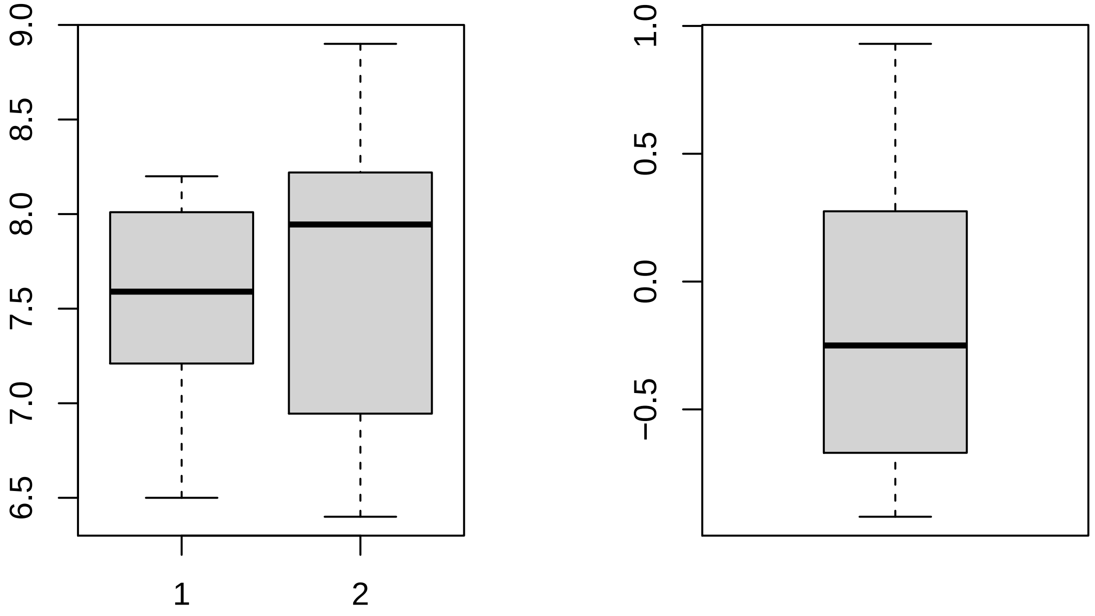
```
t.test(energy_data_before - energy_data_after)
```

```
##
##  One Sample t-test
##
## data:  energy_data_before - energy_data_after
## t = 1.6538, df = 11, p-value = 0.1264
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.05101059  0.35934392
## sample estimates:
## mean of x
## 0.1541667
```

```
lemo_data_before = data_4$before[data_4[,3] == "lemo"]
lemo_data_after = data_4$after[data_4[,3] == "lemo"]
par(mfrow=c(1,2))
hist(lemo_data_before)
hist(lemo_data_after)
```



```
par(mfrow=c(1,2))
boxplot(lemo_data_before, lemo_data_after)
boxplot(lemo_data_before - lemo_data_after)
```



```
t.test(lemo_data_before, lemo_data_after, paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  lemo_data_before and lemo_data_after
## t = -0.80596, df = 11, p-value = 0.4373
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5409781  0.2509781
## sample estimates:
## mean of the differences
##                 -0.145
```

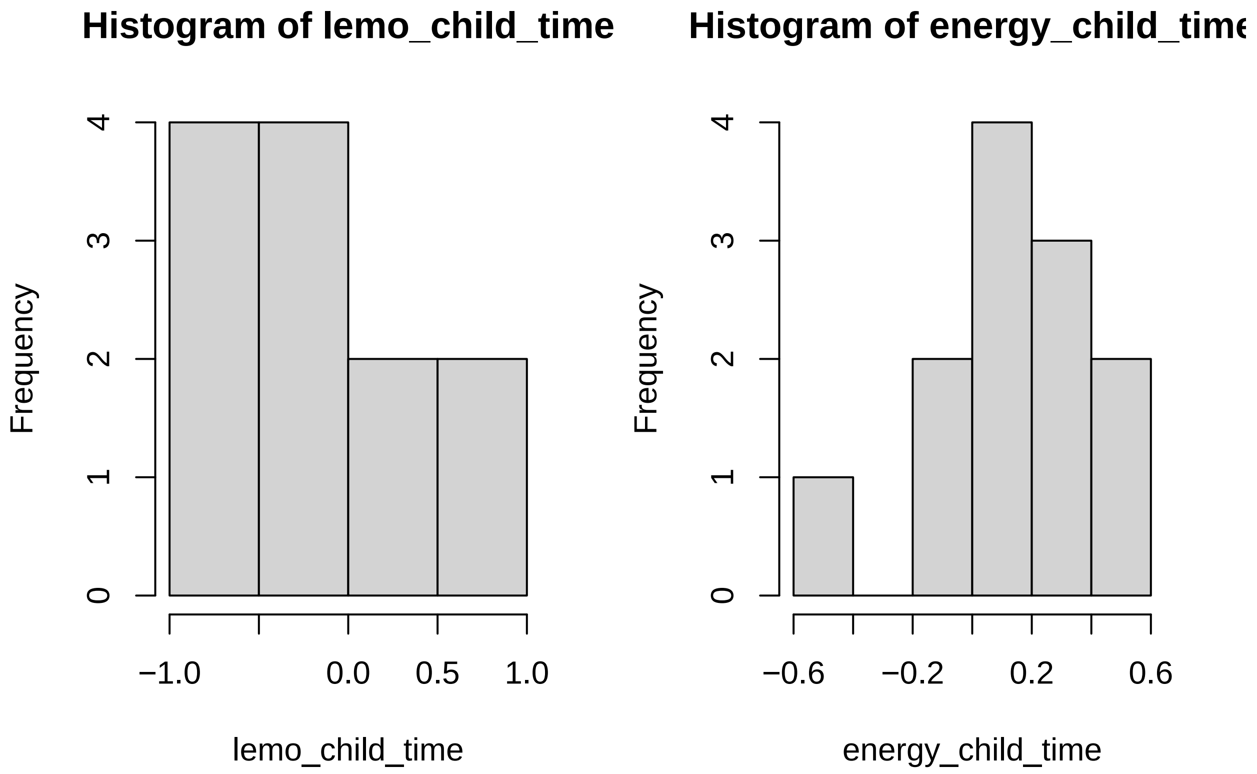
```
t.test(lemo_data_before - lemo_data_after)
```

```
##
##  One Sample t-test
##
## data:  lemo_data_before - lemo_data_after
## t = -0.80596, df = 11, p-value = 0.4373
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.5409781  0.2509781
## sample estimates:
## mean of x
##    -0.145
```

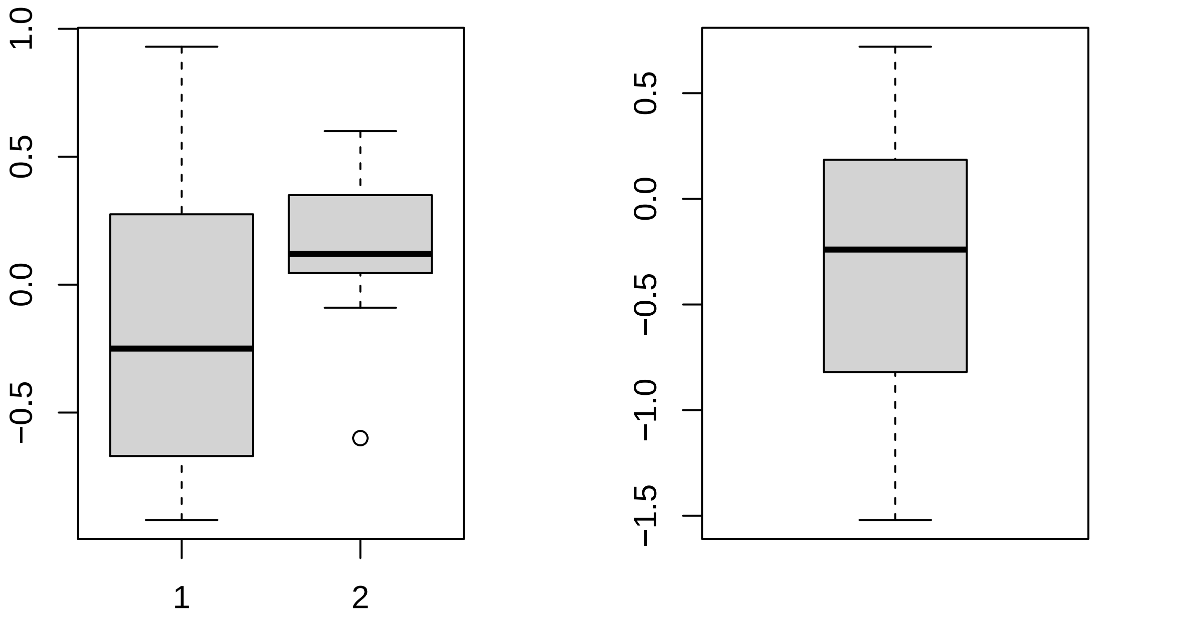
c) For each pupil compute the time difference between the two running tasks. Test whether these time differences are effected by the type of drink.

```
child_time = numeric(length(data_4$before))
for(i in 1:length(data_4$before)){
  child_time[i] =  data_4$before[i] - data_4$after[i]
}
```

```
lemo_child_time = child_time[data_4[,3] == "lemo"]
energy_child_time = child_time[data_4[,3] == "energy"]

par(mfrow=c(1,2))
hist(lemo_child_time)
hist(energy_child_time)
```



```
par(mfrow=c(1,2))
boxplot(lemo_child_time, energy_child_time)
boxplot(lemo_child_time - energy_child_time)
```



```
t.test(lemo_child_time, energy_child_time, paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  lemo_child_time and energy_child_time
## t = -1.3977, df = 11, p-value = 0.1898
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.7702717  0.1719384
## sample estimates:
## mean of the differences
##              -0.2991667
```

```
t.test(lemo_child_time - energy_child_time)
```

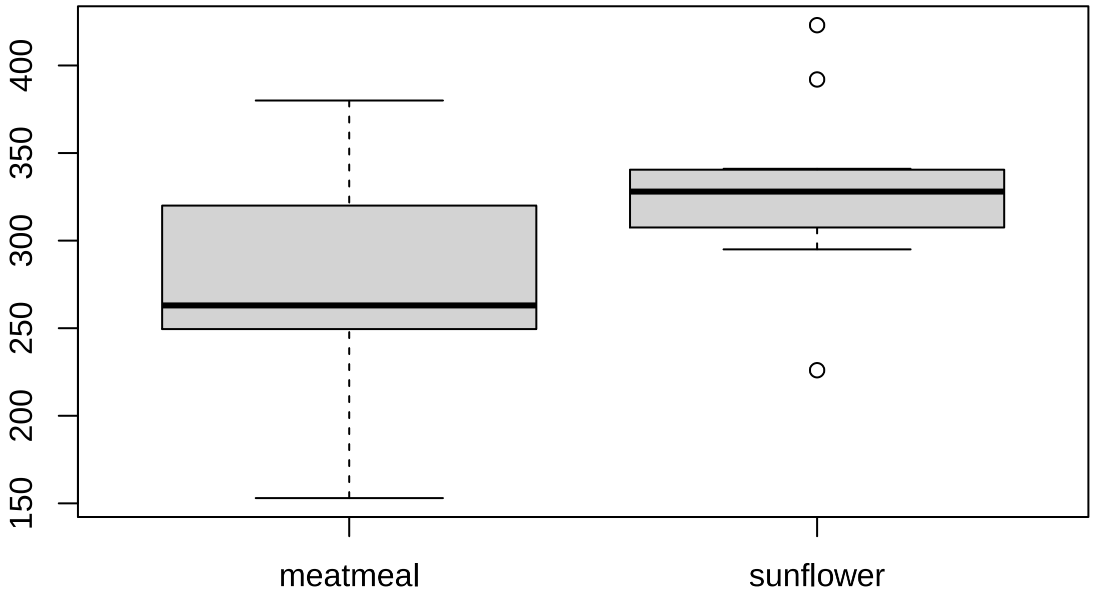
```
##
##  One Sample t-test
##
## data:  lemo_child_time - energy_child_time
## t = -1.3977, df = 11, p-value = 0.1898
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.7702717  0.1719384
## sample estimates:
##  mean of x
## -0.2991667
```

Exercise 5. *Chick weights*

The dataset chickwts is a data frame included in the standard R installation, to view it, type chickwts at the R prompt. This data frame contains 71 observations on newly-hatched chicks which were randomly allocated among six groups. Each group was given a different feed supplement for six weeks, after which their weight (in grams) was measured. The data frame consists of a numeric column giving the weights, and a factor column giving the name of the feed supplement.

a) Test whether the distributions of the chicken weights for meatmeal and sunflower groups are different by performing three tests: the two samples t-test (argue whether the data are paired or not), the Mann-Whitney test and the Kolmogorov-Smirnov test. Comment on your findings.

```
data("chickwts")
meatmeal = chickwts$weight[chickwts$feed == "meatmeal"]
sunflower = chickwts$weight[chickwts$feed == "sunflower"]
boxplot(meatmeal, sunflower, names=c("meatmeal", "sunflower"))
```



```
t.test(meatmeal, sunflower, paired=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  meatmeal and sunflower
## t = -2.1564, df = 18.535, p-value = 0.04441
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -102.572435   -1.442716
## sample estimates:
## mean of x mean of y
##  276.9091  328.9167
```

Since the Welch Two Sample t-test has p-value=0.04441 < 0.05, it is the case to reject H_0. The conclusion should be there exist certain difference between the meatmeal and sunflower. Additionally, the data are not paired, because if we set "paired=TRUE", we would get an error message- "not all arguments have the same length".

```
wilcox.test(meatmeal, sunflower)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  meatmeal and sunflower
## W = 36, p-value = 0.06882
## alternative hypothesis: true location shift is not equal to 0
```

Mann-Whitney test has a p-value $= 0.06882 > 0.05$, so the conclusion is that there is no such significant difference between two groups.

```
ks.test(meatmeal,sunflower)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  meatmeal and sunflower
## D = 0.47727, p-value = 0.1085
## alternative hypothesis: two-sided
```

Kolmogorov-Smirnov test has a p-value $= 0.1085 < 0.05$. From this result we can conclude that the two populations are extremely unsymmetrical in shapes.

c) Conduct a one-way ANOVA to determine whether the type of feed supplement has an effect on the weight of the chicks. Give the estimated chick weights for each of the six feed supplements. What is the best feed supplement?

```
chickaov=lm(weight ~ feed, data=chickwts)
anova(chickaov)
```

```
## Analysis of Variance Table
##
## Response: weight
##           Df Sum Sq Mean Sq F value    Pr(>F)
## feed       5 231129   46226  15.365 5.936e-10 ***
## Residuals 65 195556    3009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By conducting the one-way ANOVA, it is easy to see that the p-value $=$ 5.936e-10 < 0.05, which tells that there exist at least such a type has an extremely different average weight.

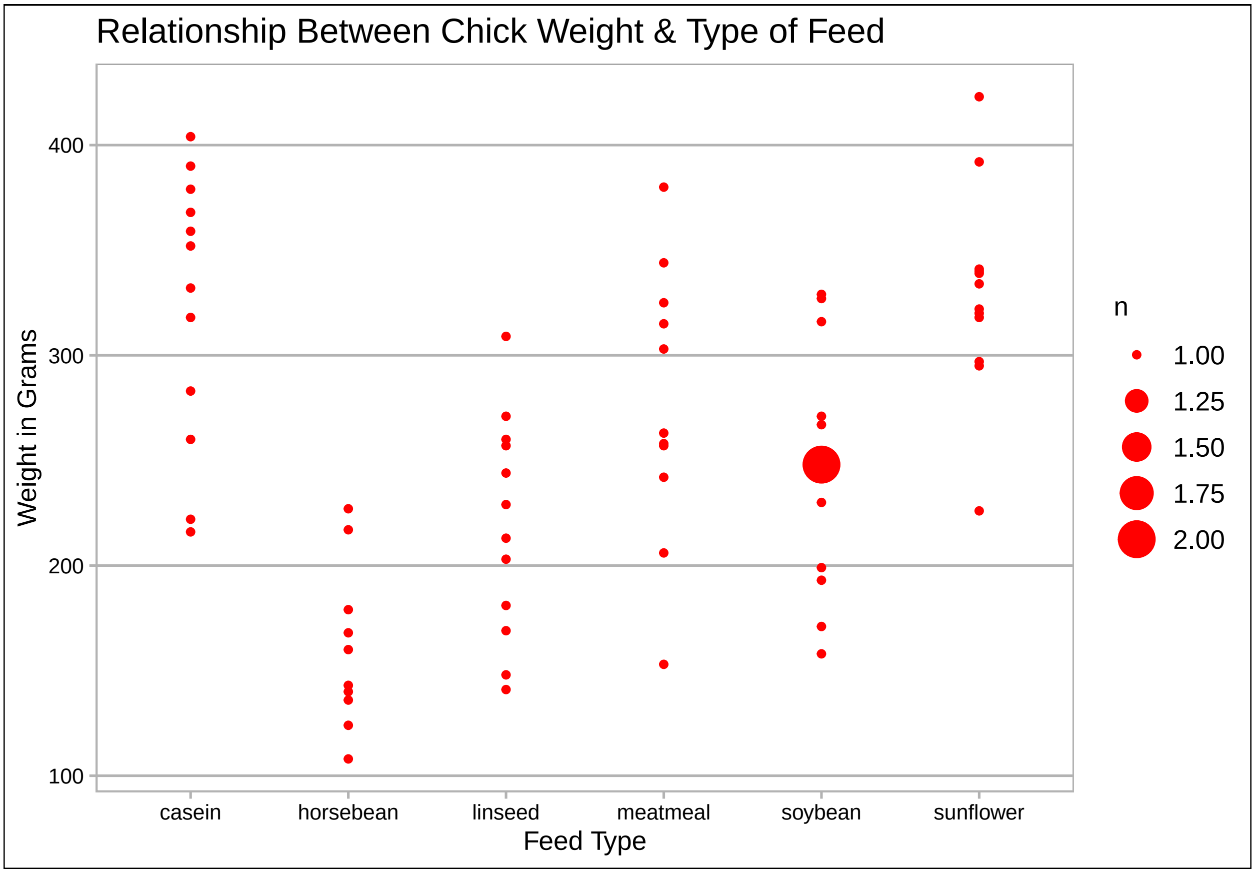
```
library(ggplot2)
library(ggthemes)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
cdata = tbl_df(chickwts)
```

```
## Warning: `tbl_df()` was deprecated in dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
```

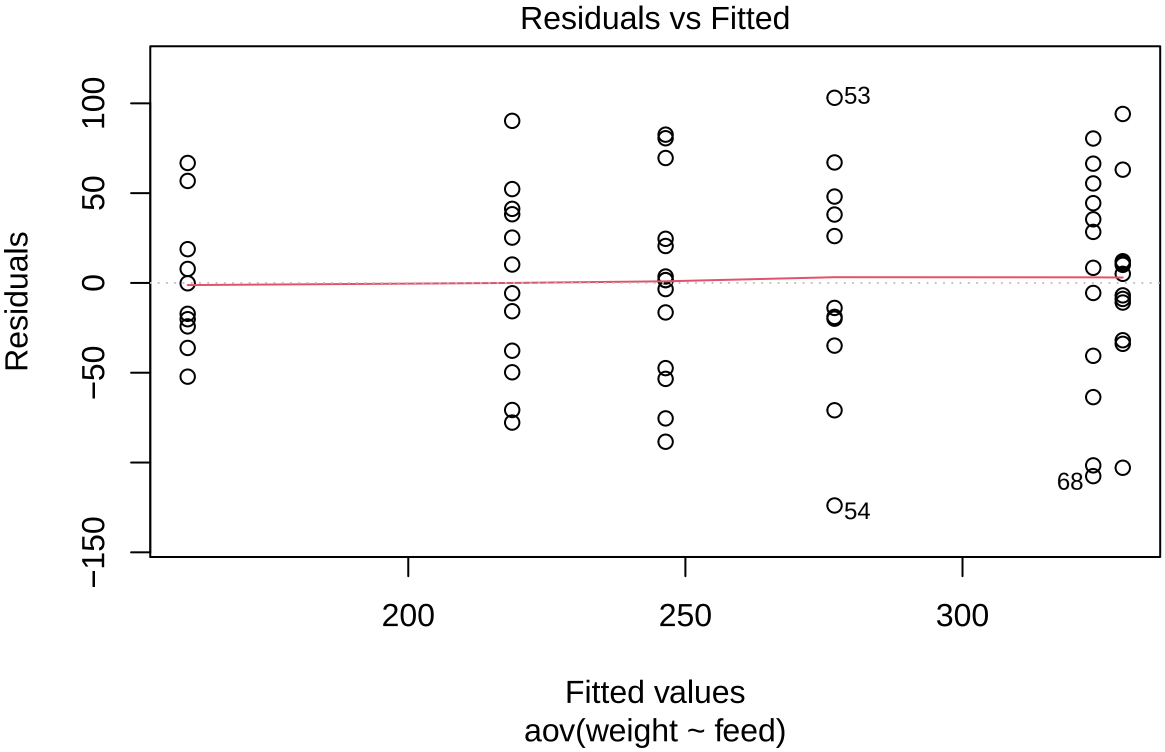
```
cdata.wt.feed = cdata %>%
  ggplot(aes(x=feed, y=weight)) +
  geom_count(color="red") +
  labs(title="Relationship Between Chick Weight & Type of Feed",
       x="Feed Type", y="Weight in Grams") +
  theme_calc()
cdata.wt.feed
```

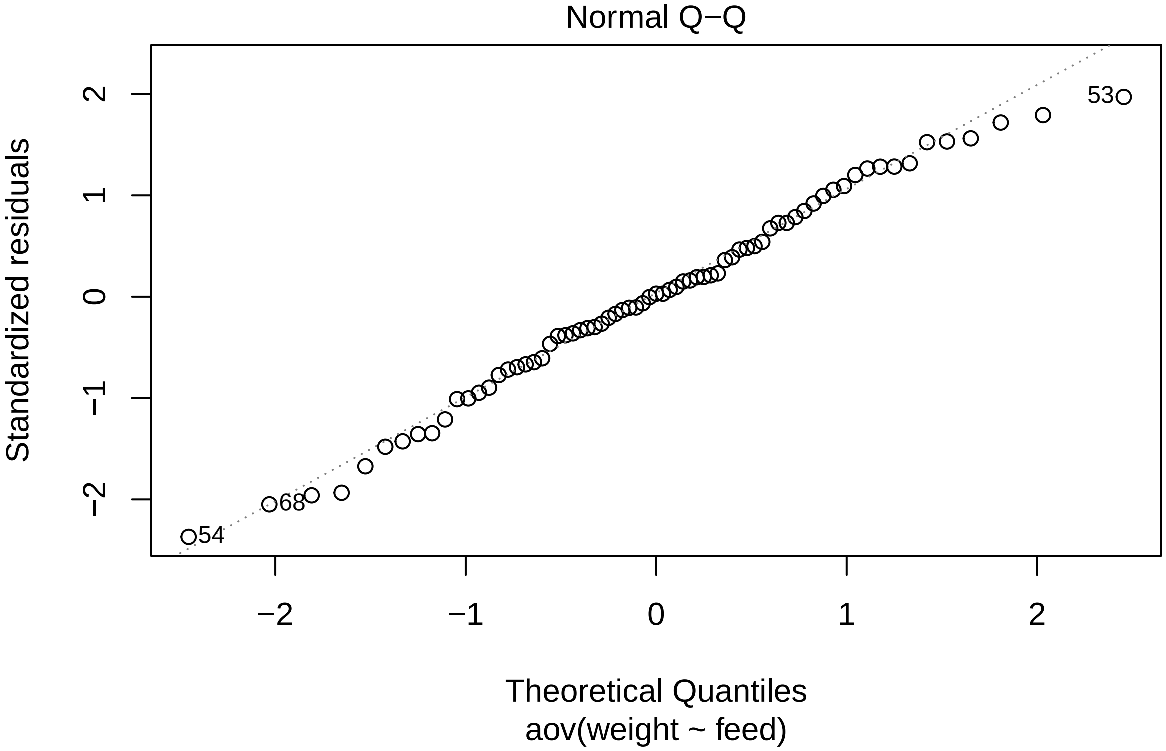


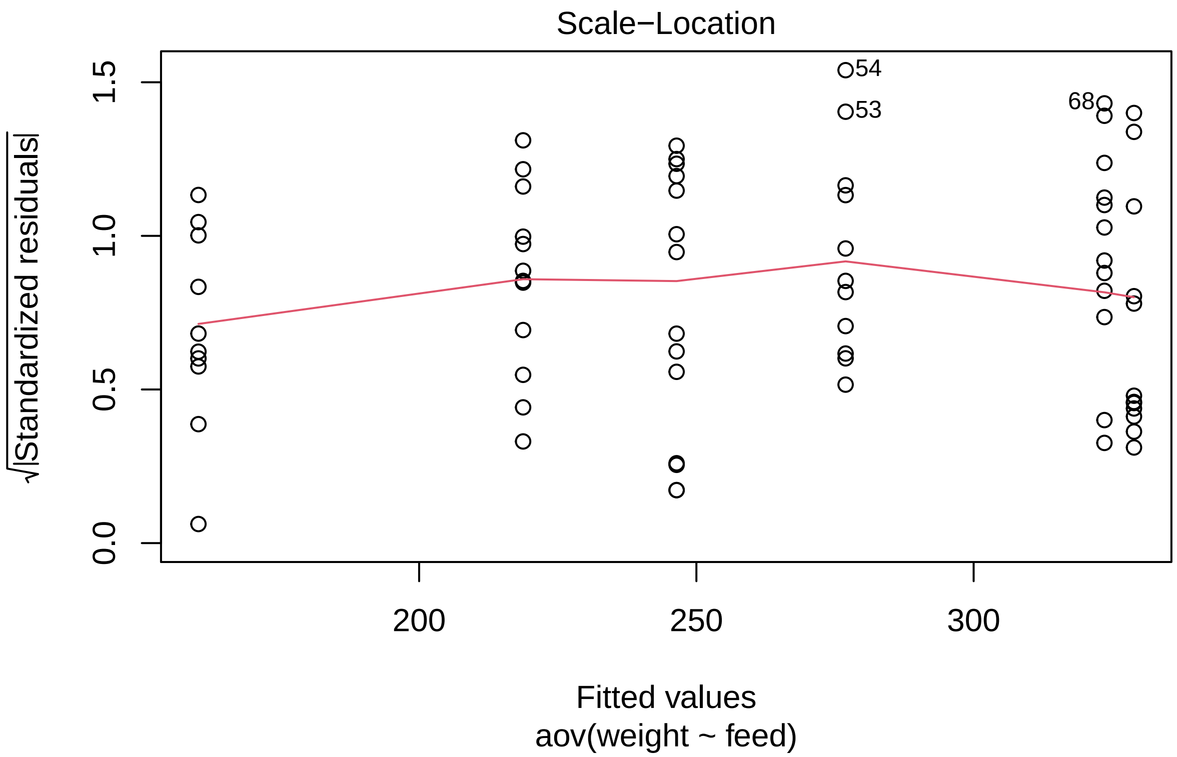
The above plot shows that the casein and sunflower are the best feed supplement.

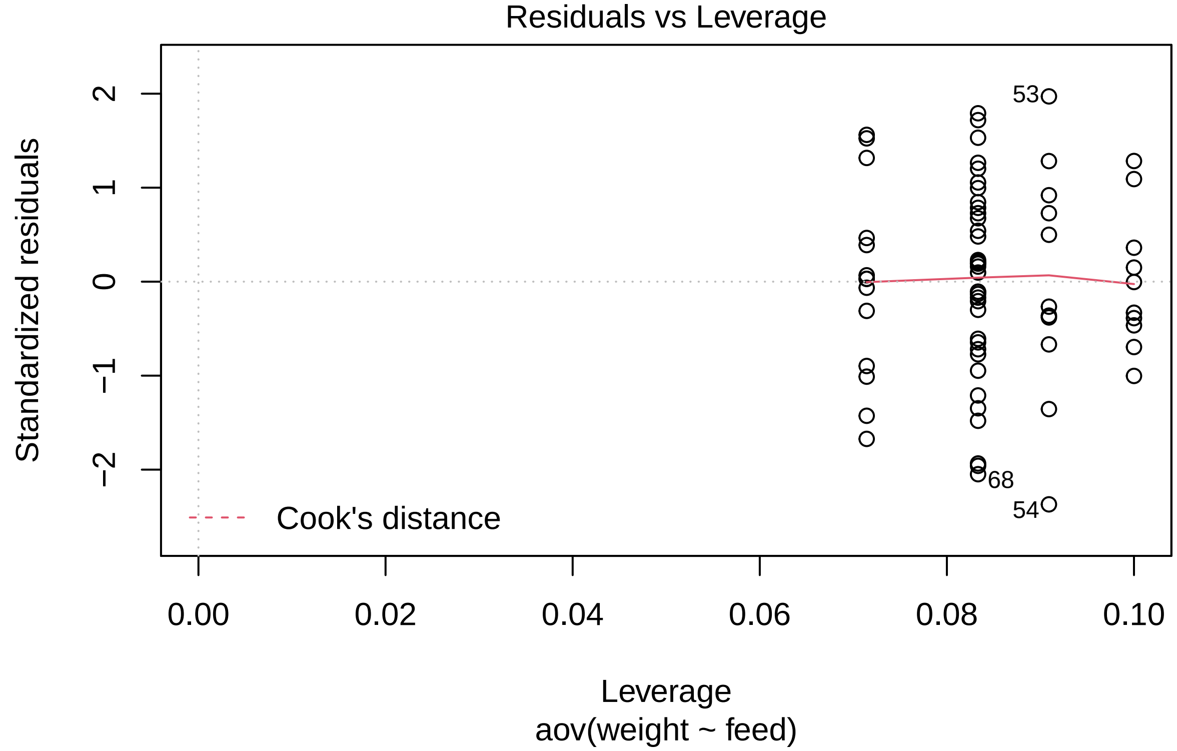
c) Check the ANOVA model assumptions by using relevant diagnostic tools.

```
caov = aov(weight~feed, data=chickwts)
plot(caov)
```


Residuals vs Fitted
Residuals
100
50
0
−50
−150
200
250
300
53
54
68
Fitted values
aov(weight ~ feed)


Normal Q−Q
Standardized residuals
2
1
0
−1
−2
−2
−1
0
1
2
53
68
54
Theoretical Quantiles
aov(weight ~ feed)





These plots shows that the ANOVA model assumptions is in line with expectations. In the case like this, we can conclude that values have equal variance.

```
summary(caov)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## feed         5 231129   46226   15.37 5.94e-10 ***
## Residuals   65 195556    3009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Additionally, since the p-value < 0.05, it also shows the normality of the values.

d) Does the Kruskal-Wallis test arrive at the same conclusion about the effect of feed supplement as the test in b)? Explain possible differences between conclusions of the Kruskal-Wallis and ANOVA tests.

```r
kruskal.test(weight ~ feed, data=chickwts)
```

```
## 
##  Kruskal-Wallis rank sum test
## 
## data:  weight by feed
## Kruskal-Wallis chi-squared = 37.343, df = 5, p-value = 5.113e-07
```

Kruskal-Wallis test has a p-value $=$ 5.113e-07 < 0.05, which means there exist such a feed influenced the weight. And the one-way ANOVA also shows the same conclusion. On the other hand, Kruskal-Wallis test is based on ranks instead of normality as ANOVA. It means that the ANOVA tested the normality of values from means, but Kruskal-Wallis tested on comparison of the ranks of the means.