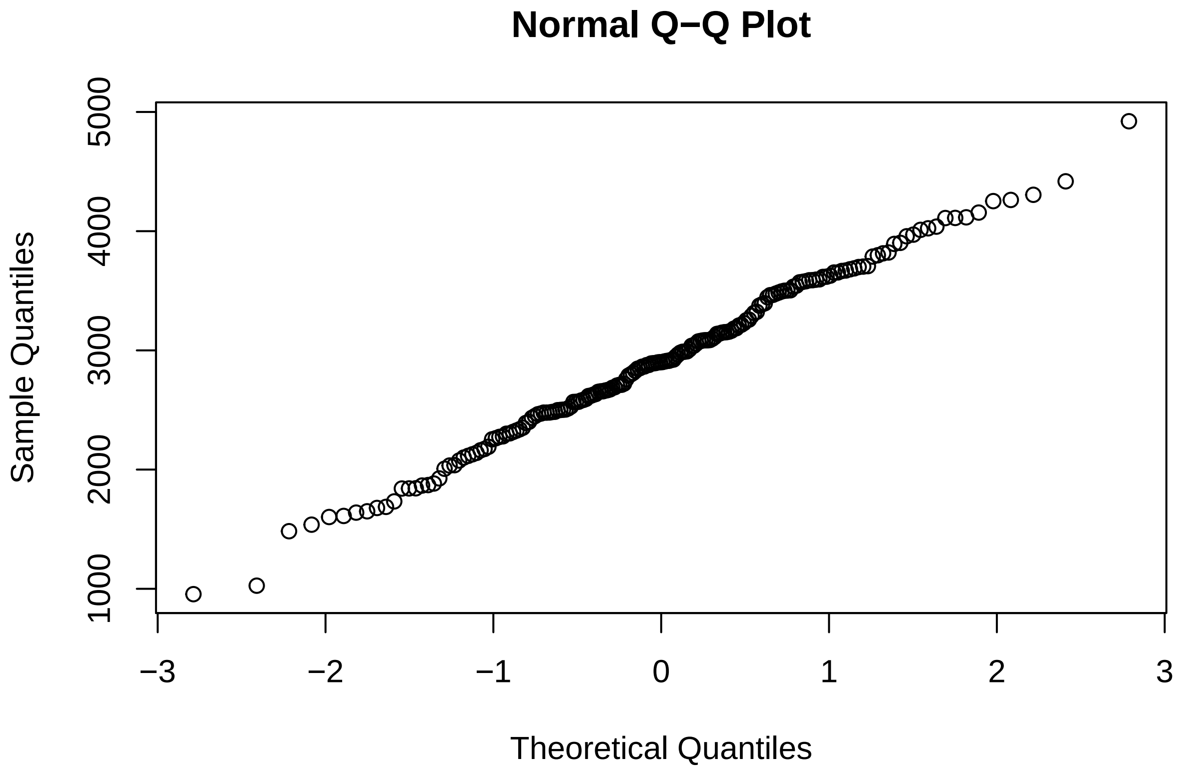
Assignment 1

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

2021/2/19

Exercise 1. ***Birthweights***

```
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 points in 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
```

The value of mu is equal to 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
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.

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

b)

```
##
##  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 the set threshold, then the null hypothesis $H0$ is rejected and $H1$ fails to be rejected. In this case, the mean of the data is larger than 2800. The same result can be obtained if *alpha*>=0.013, because the p-value is equal with 0.013. If *alpha* has a value that is lower than 0.013 then we retain the null hypothesis $H0$.

c) The confidence interval found for point a) is calculated for a confidence interval of 90%, while the interval that is calculated when performing a t-test is 95% by default. In the case of the interval obtained at point b), one side of the interval is not defined because we are interested only in the values that are greater than our mean and we don't take into consideration if a value is smaller than our mean. Because of this the confidence interval, calculated at point b), the interval will be from the 10% quantile to infinity. This effect will also make the lower bound of the interval to be equal with the lower bound of the interval calculated at point a), even though they are calculated for different confidence intervals(90%, 95% respectively).

This explanation can be tested simply by creating two t-tests, one with the default confidence level and one with a confidence level of 90%.

```
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
```

```
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
```

After realizing the two tests we can observe that the values of the confidence interval are different from one another. Another observation is that in comparison to point b), the interval is defined on both sides, because in this case, we are interested in values on both sides on the mean, not just bigger.

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

```
n = m = 30#Initialize the variables that are going to be used to calculate the power
mu = 180
nu = 175
sd = 5
B = 1000
p = numeric(B)
for(b in 1:B){#Calculate the p-value for a "B" number of times
  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: 0.962
```

```
nu = seq(175, 185, by=0.25)#Initialize the variables that are going to be used to calculate the power
power_a = numeric(length(nu))
for(i in 1:length(nu)){#Calculate the power for each "nu" value
  for(b in 1:B){
    x = rnorm(n, mu, sd)
    y = rnorm(m, nu[i], sd)
    p[b] = t.test(x, y, var.equal=TRUE)[[3]]
  }
  power_a[i] = mean(p<0.05)#Calculate the power function of a particular value of "nu"
}
```

a)

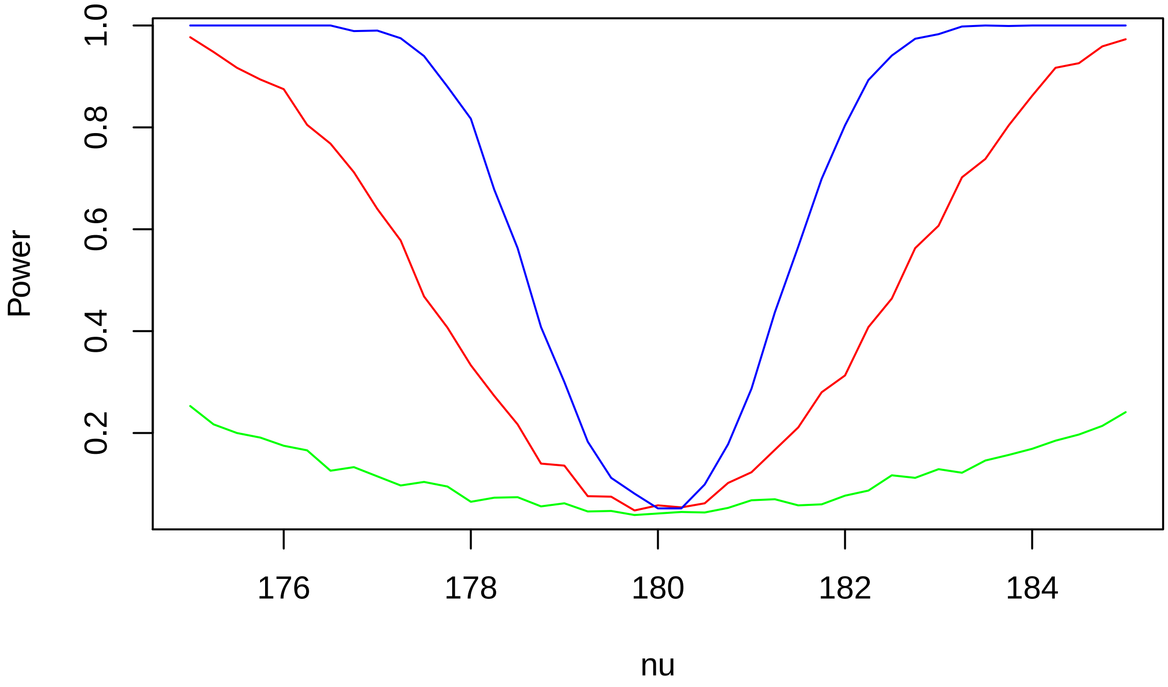
```
n = m = 100#Initialize the variables that are going to be used to calculate the power
nu = seq(175, 185, by=0.25)
power_b = numeric(length(nu))
for(i in 1:length(nu)){
  for(b in 1:B){
    x = rnorm(n, mu, sd)
    y = rnorm(m, nu[i], sd)
    p[b] = t.test(x, y, var.equal=TRUE)[[3]]
  }
  power_b[i] = mean(p<0.05)
}
```

b)

```

n = m = 30#Initialize the variables that are going to be used to calculate the power
sd = 15
nu = seq(175, 185, by=0.25)
power_c = numeric(length(nu))
for(i in 1:length(nu)){
  for(b in 1:B){
    x = rnorm(n, mu, sd)
    y = rnorm(m, nu[i], sd)
    p[b] = t.test(x, y, var.equal=TRUE)[[3]]
  }
  power_c[i] = mean(p<0.05)
}
```

c)

```

plot(nu, power_a, type="l", col="red", ylab="Power", xlab="nu")
points(nu, power_b, type="l", col="blue")
points(nu, power_c, type="l", col="green")
legend(10,90, legend=c("Power a", "Power b", "Power c"), col=c("red", "blue", "green"))
```



d)

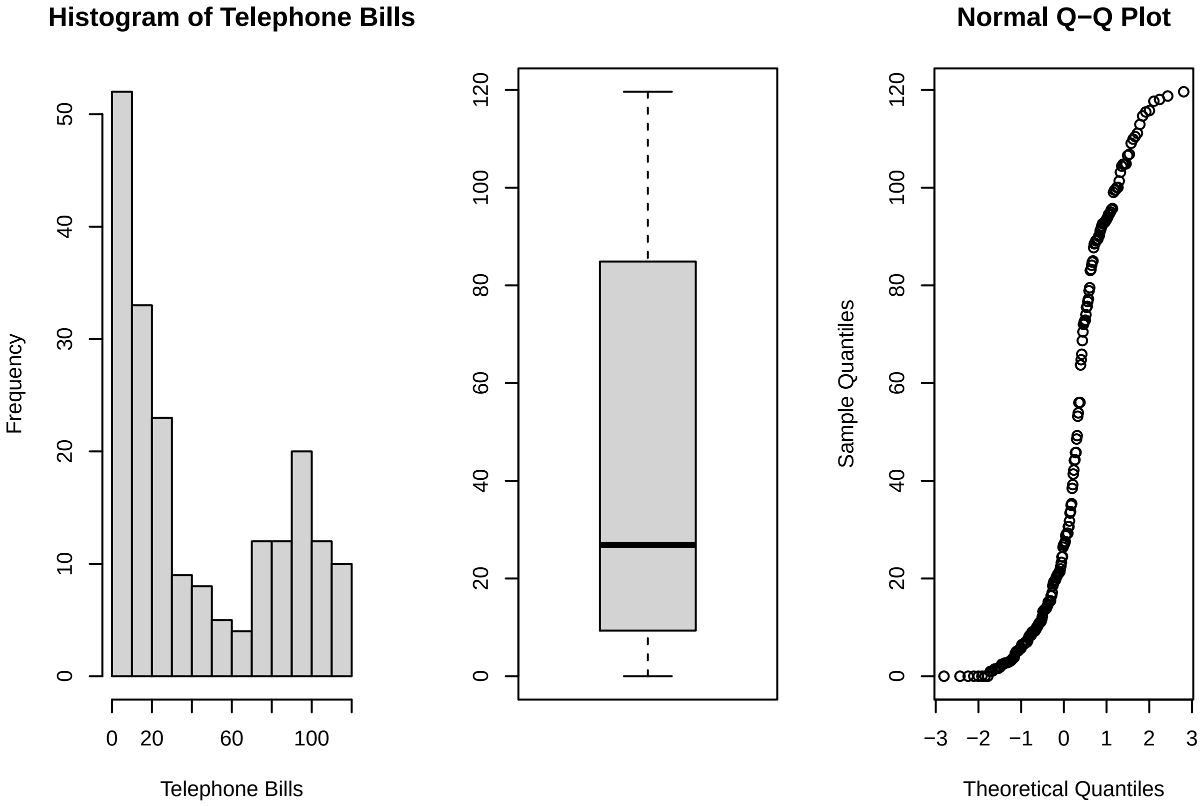
- The first observation illustrated in the plot, is that the power has a higher value as *nu* value get closer to the extremities and is equal with 0 when *nu* is equal with 180, this applies for all three cases.
- Another observation is that all three lines have the shape of a bell curve but upside down.
- If the red line is considered the base case, if we increase the number of samples, the blue line is obtained.

In this case, the extremities get closer to 1 much faster. This indicates a higher power, so as it was expected: *increasing the sampling size increases the power level.*

- In the case of the green line, we can observe the opposite. We have drop in power level compared to the red line. In conclusion *increasing the standard deviation will decrease the power function.*

Exercise 3. ***Telecommunication company***

```
telephone = read.table('telephone.txt', header=T)
med = median(telephone$Bills)
par(mfrow = c(1,3))
hist(telephone$Bills, main='Histogram of Telephone Bills', xlab='Telephone Bills')
#inconsistency in shape
boxplot(telephone$Bills, xlab='')
qqnorm(telephone$Bills)
```

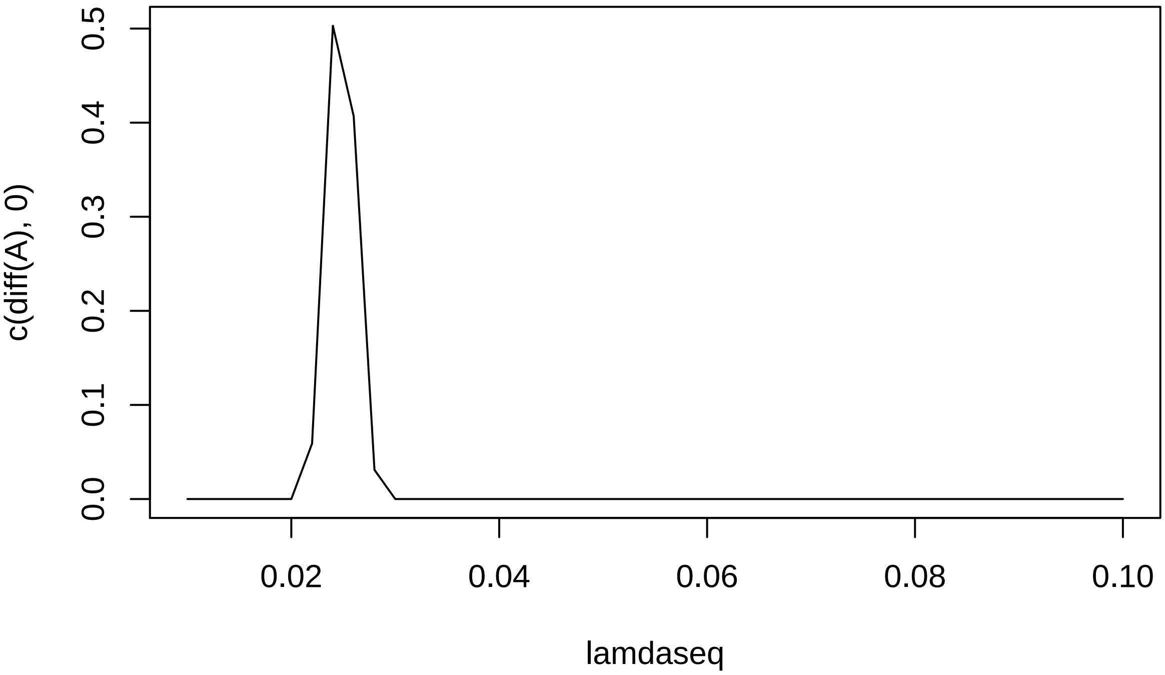


a)

Considering the sinusoidal form of the Q-Q plot of the Theoretical Quantiles and the inconsistency in shape observed from the histogram, it is safe to assume that the Bills data does not follow a normal distribution. An advice for the marketing manager would be to try to increase the plan price for the customers that have very low bills maybe by offering extra data or other bundles.

```
mfrow = c(1,1)
lamdaseq = seq(0.01, 0.1, by = 0.002)
A = numeric(length(lamdaseq))

for (i in 1:length(lamdaseq))
  {B = numeric(1000)
```

```
   lambda = lamdaseq[i]
   for (j in 1:length(B))
     {sample = rexp(1000, lambda)
      B[j] = median(sample)
      }
  A[i] = (length(B[B<med])/length(B)) #Building a vector with the values for which B<med, divided by th
}

plot(lamdaseq, c(diff(A), 0), type='l') #for this plot, a random sample has been drawn
```



b)

```
#with the lambda values from lambdaseq. For each sample, 1000 observations were drawn and compared to t
```

Considering the given interval, it can be concluded that the median point is found between 0.02 and 0.03.

```
B=1000
Tstar=numeric(B)
for(i in 1:B) {
  Xstar=sample(telephone$Bills,replace=TRUE)
  Tstar[i]=median(Xstar)}
Tstar25 = quantile(Tstar,0.025)# construct 95% confidence interval
Tstar975 = quantile(Tstar,0.975)
ciTstar = c(2*median(Tstar)-Tstar975, 2*median(Tstar)-Tstar25)
print(median(Tstar))#The obtained 95% CI for the population median of the given sample is [0.20, 0.32],
```

c)

```
## [1] 26.905
```

```
B = numeric(1000)
for (j in 1:length(B)){
  sample = rexp(200, 0.026)
  B[j] = median(sample)
}
B25 = quantile(B, 0.025)
B975 = quantile(B, 0.975)
med = median(B)
print(med)
```

d)

```
## [1] 26.54772
```

```
ciB = c(2*med - B975, 2*med - B25)
print(ciB)
```

```
##    97.5%     2.5%
## 20.80586 31.89706
```

```
#The obtained 95% confidence interval for the population median is [21.69263, 32.15699], around the mea
```

```

pl = sum(Tstar<40)/B
pr = sum(Tstar>=40)/B
p = 2*(min(pl,pr))
p
```

e)

```
## [1] 0.6722096
```

```
#The bootstrap p-value is 0, which means that the data from the t-test can not not be trusted, hence th

# test percentage bill <10 at most 25% from surrogate data sample
B =1000
Tstar=numeric(B)
for(i in 1:B) {
  Xstar=sample(telephone$Bills,replace=TRUE)
  Tstar[i]=mean(Xstar<10)
}
pl = sum(Tstar<0.25)/B
pr = sum(Tstar>=0.25)/B
p = 2*(min(pl,pr))
mean(Tstar)
```

```
## [1] 0.26205
```

```
pl;pr;p
```

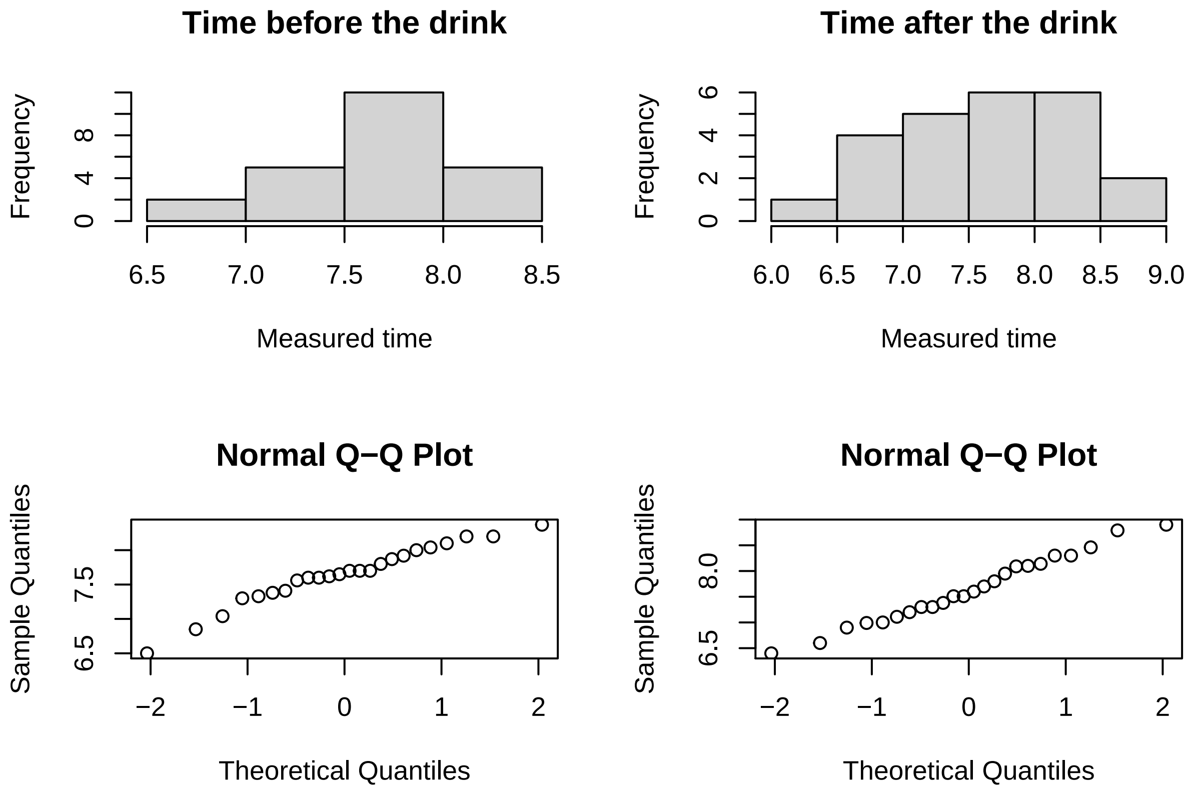
```
## [1] 0.32

## [1] 0.68

## [1] 0.64
```

```
#The obtained p-value from the test is 0.672, hence H0 is validated, confirming there is a significant
```

Exercise 4. *Energy drink*

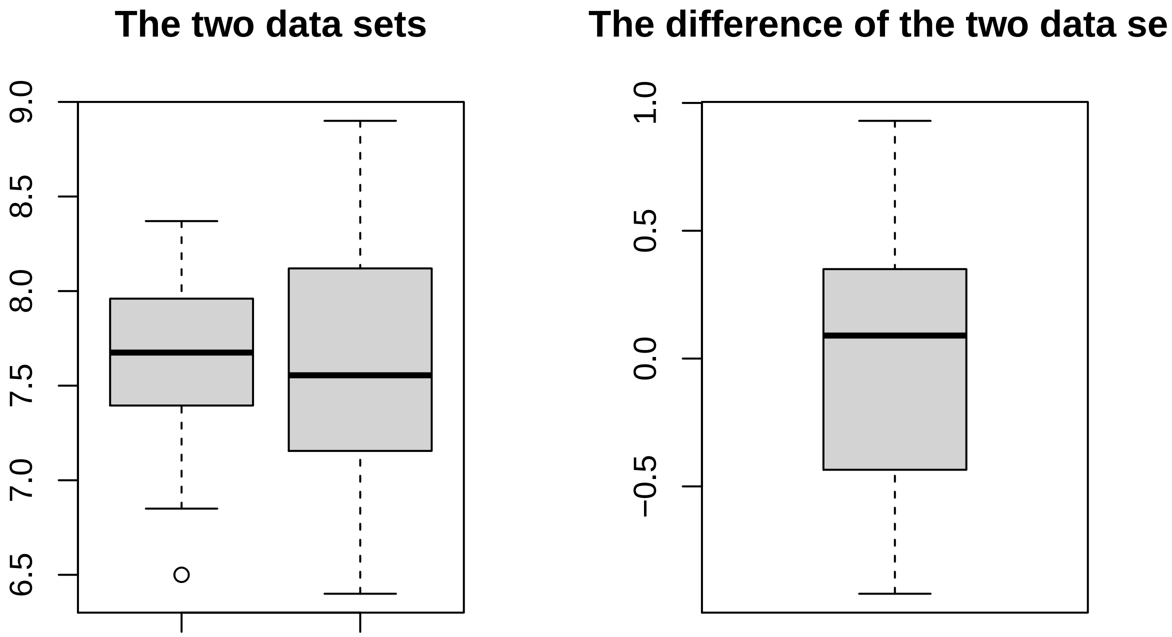
a) The first step is to create a histogram and a QQ-plot for the two data sets to verify normality of distribution.

```
data_4 = read.table(file="run.txt", header=TRUE)#Read the data from run.txt
par(mfrow=c(2,2))
hist(data_4$before, main="Time before the drink", xlab="Measured time")#Creates a histogram from the da
hist(data_4$after, main="Time after the drink", xlab="Measured time")#Creates a histogram from the data
qqnorm(data_4$before)#Creates a QQ-norm from the time before before the drink
qqnorm(data_4$after)#Creates a QQ-norm from the time before after the drink
```



After looking at the histograms and the QQ-plots, it can be assumed that the distribution of the data is normal.

```
par(mfrow=c(1,2))
boxplot(data_4$before, data_4$after, main="The two data sets")#Creates a boxplot from the data with the
boxplot(data_4$before - data_4$after, main="The difference of the two data sets")#Creates a boxplot fro
```



The next step is to analyze the data using a box plot. If we look at the three box plots we can observe a small variation between the measured time before and after drinking the energy drink. This could mean that the energy drink didn't have any real effect on the children, we can also test this by performing a two sided t-test on the obtained data and see if the p-value > 0.05.

```
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
```

After performing the t-test in two different ways and getting the same result we can conclude that we fail to reject the null hypothesis $H0$.

```
energy_data_before = data_4$before[data_4[,3] == "energy"]
energy_data_after = data_4$after[data_4[,3] == "energy"]
par(mfrow=c(2,2))
#hist(energy_data_before, main="Run time before energy drink", xlab="Measured time")
#hist(energy_data_after, main="Run time after energy drink", xlab="Measured time")
#boxplot(energy_data_before, energy_data_after, main="The two data sets")
#boxplot(energy_data_before - energy_data_after, main="The difference of the two data sets")
```

b) The next step is to analyze the data concerning the energy drinks. If we look at the graphs we an observe that there is little difference between the measured time before and after drinking the energy drink. This could mean that the energy drink didn't have any real effect on the children, we can also test by performing a t-test on the obtained data and see if the p-value is not significant.

```
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
```

After performing the two t-tests and obtaining the same p-value from both of them, we can conclude that we fail to reject $H0$.

```
lemo_data_before = data_4$before[data_4[,3] == "lemo"]
lemo_data_after = data_4$after[data_4[,3] == "lemo"]
par(mfrow=c(2,2))
#hist(lemo_data_before, main="Run time before placebo", xlab="Measured time")
#hist(lemo_data_after, main="Run time after placebo", xlab="Measured time")
#boxplot(lemo_data_before, lemo_data_after, main="The two data sets")
#boxplot(lemo_data_before - lemo_data_after, main="The difference of the two data sets")
```

We could also try to analyze the data obtained from the children that took the placebo. If we look at the graphs, we an observe that the difference between the the measured time before and after drinking is very small. This is expected from the placebo, this can also be tested by performing a t-test on the obtained data and see if there is a p-value not significant.

```
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
```

After performing the two t-tests and obtaining the same p-value from both of them we can get to the conclusion that we fail to reject the null hypothesis $H0$.

```

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(2,2))
#hist(lemo_child_time, main="Run time placebo", xlab="Measured time")
#hist(energy_child_time, main="Run time energy drink", xlab="Measured time")
#boxplot(lemo_child_time, energy_child_time, main="The two data sets")
#boxplot(lemo_child_time - energy_child_time, main="The difference of the two data sets")
```

c) The last method to see if there is any difference between the energy drink and the placebo is to test each child individually and compare the results. After looking at the histograms and the three box plot we can observe that the difference between the two data sets is very small. Due to this, we can assume assume that there is no difference between them, but we have to test this using the t-test and see if the p-values > 0.05.

```
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
```

After performing the two t-tests and obtaining the same p-value from both of them we can get to the conclusion that we fail to reject the null hypothesis $H0$.

d) Yes, there are two improvement that can be applied for design experiment for point b) and c). The first one is to implement *crossover experimental design*, in this way we can eliminate bias. The second improvement is to increase the sample size so that we can have a better representation of the two distributions.

Exercise 5. ***Chick weights***

```
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)
```

a)

```
##
##  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.

```

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

b)

```
## 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 $= 5.936\text{e-}10 < 0.05$, which tells that there exist at least such a type has an extremely different average weight.

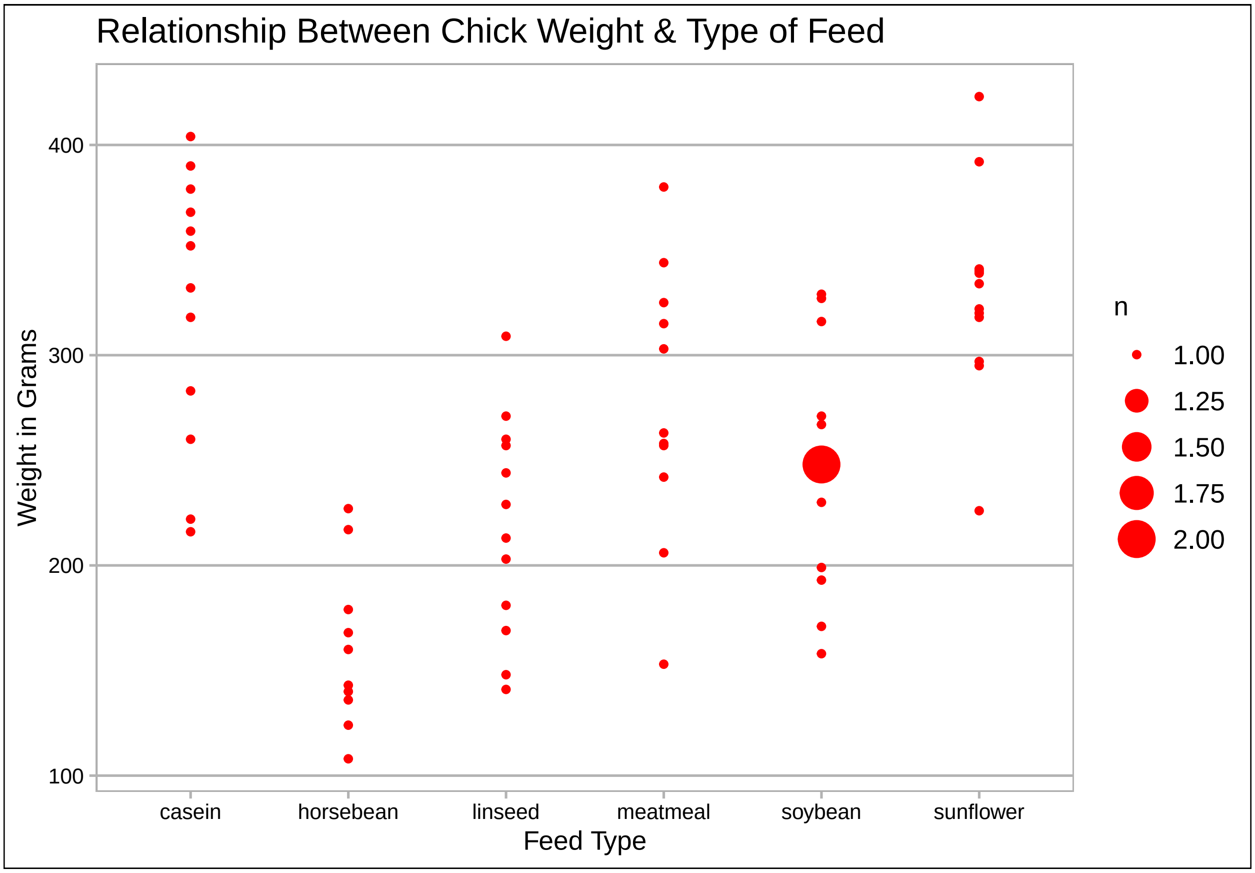
```
library(ggplot2)
library(ggthemes)
library(dplyr)
```

```
##
## 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)
```

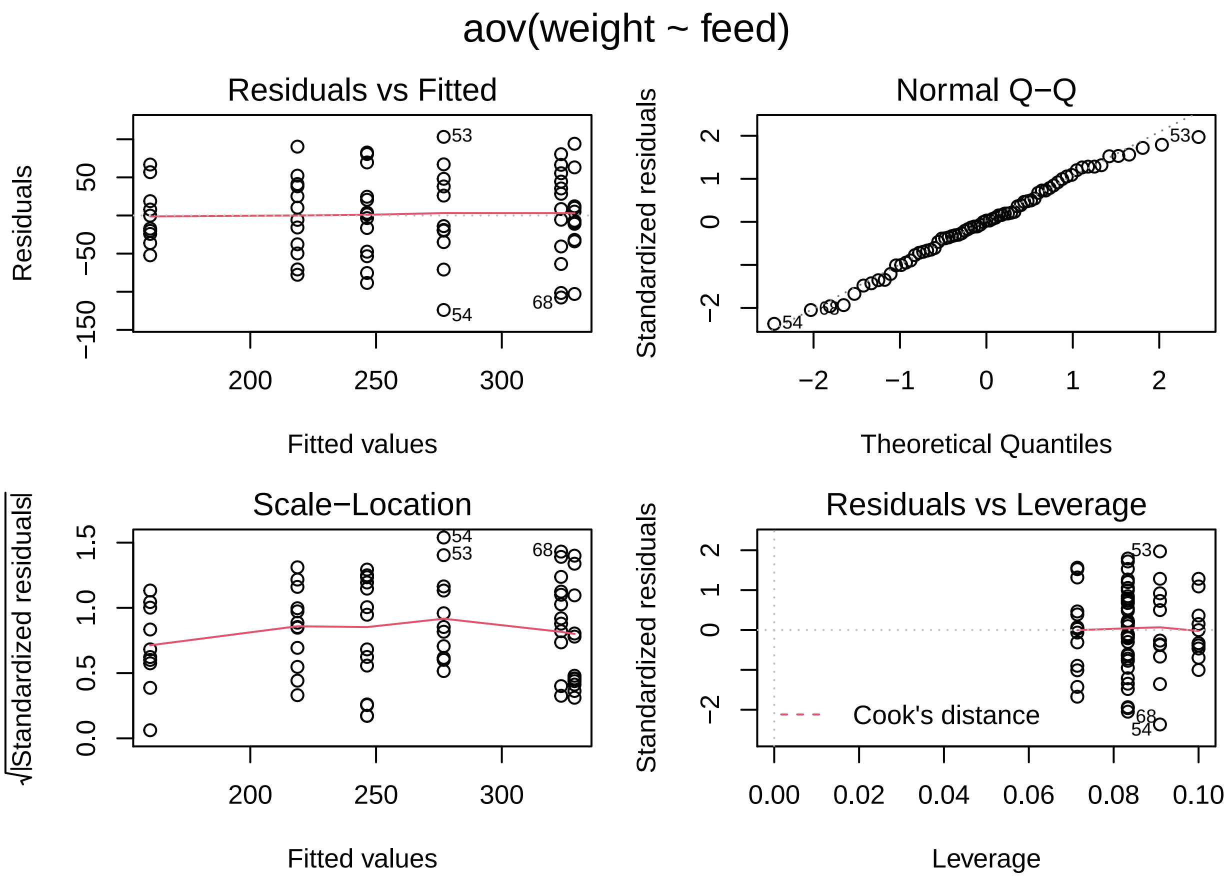
```
## Warning: `tbl_df()` was deprecated in dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
```

```
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.

```
caov = aov(weight~feed, data=chickwts)
opar = par(mfrow = c(2, 2), oma = c(0, 0, 1.1, 0),
           mar = c(4.1, 4.1, 2.1, 1.1))
plot(caov)
```



c)

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.

```

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

d)

```
## 
##  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 $= 5.113\text{e-}07 < 0.05$, 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.