# EDDA ASSIGNMENT 1

Ella Smorenburg (2618639), Roel Rotteveel(271547), Yoes Ywema(271544)

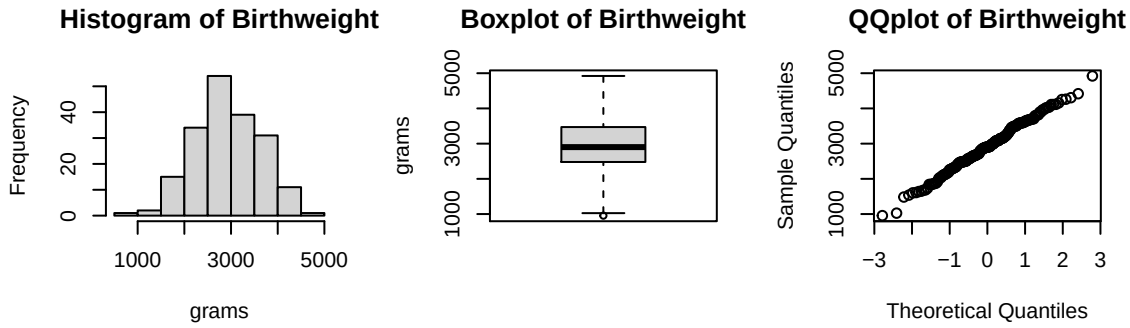
```
#install.packages('pander')
library(pander)
#panderOptions('round', 2)
#panderOptions('keep.trailing.zeros', TRUE)
options(digits=3)
```

## Exercise 1. Birth weights

```
# setwd("~/Folder with datasets")
data = read.table("birthweight.txt", header=TRUE)
birthweight = data
```

### a

```
plot3 <-function(x, title, xaxishist){
  par(mfrow=c(1,3))
  hist(x,main=paste0("Histogram of ", title),xlab=xaxishist)
  boxplot(x, main=paste0("Boxplot of ", title), ylab=xaxishist)
  qqnorm(x, main=paste0("QQplot of ", title))}
plot3(data$birthweight, "Birthweight", 'grams')
```




The normality of the 'Birthweight' data set is checked by observing a histogram, boxplot and a Q-Q plot. According to these figures the data seems to be normally distributed. The form of the histogram looks approximately 'Bell-shaped'. The median of the Birthweight' data set is approximately in the middle of the box of the boxplot, and the whiskers are about the same on both sides. The points from the 'Birthweight' data set form a roughly straight line in the Q-Q plot. These observations confirm that the data is normally distributed.

```r
sample_mean = mean(data$birthweight)
sample_mean
```

```
## [1] 2913
```

A point estimate for the mean can be derived by computing the sample mean (average over all sample values). The sample mean is 2913.

```r
sample_sd = sd(data$birthweight)
quantile = qnorm(0.05) # in range [0.05, 0.95]
margin_error = abs(quantile)*(sample_sd/sqrt(length(data$birthweight)))

sample_mean-margin_error #lower bound
```

```
## [1] 2830
```

```r
sample_mean+margin_error #upper bound
```

```
## [1] 2997
```

```r
# To check:
t.test(birthweight$birthweight, conf.level = 0.90)[4]
```

```
## $conf.int
## [1] 2829 2997
## attr(,"conf.level")
## [1] 0.9
```

The margin error is computed by multiplying the quantiles of a regular normal distribution at 0.05 and 0.95 with the sample standard error divided by the square root of the length of the data set (number of instances). Subtracting this margin error from the sample mean gives us the lower bound of the confidence interval. Adding this margin error to the sample mean gives us the upper bound of the confidence interval.

## b

```r
# t-test with alpha = 0.1
t.test(data$birthweight, mu=2800, alternative = "greater", conf.level = 0.90)[[3]]
```

```
## [1] 0.0136
```

```r
t.test(data$birthweight, mu=2800, alternative = "greater", conf.level = 0.90)[[4]]
```

```
## [1] 2848  Inf
## attr(,"conf.level")
## [1] 0.9
```

```r
#t-test with alpha = 0.05.
t.test(data$birthweight, mu=2800, alternative = "greater", conf.level = 0.95)[[3]]
```

```
## [1] 0.0136
```

```r
t.test(data$birthweight, mu=2800, alternative = "greater", conf.level = 0.95)[[4]]
```

```
## [1] 2829  Inf
## attr(,"conf.level")
## [1] 0.95
```

The claim of the expert is confirmed by this t-test. The hypotheses are the following:

- H0:the true mean is not greater than 2800
- H1:the true mean is greater than 2800.

The p-value for rejecting the null hypothesis is 0.014. This is independent of the value of alpha since this value is not involved in computing the t-statistics or the p-value. It can instead be used as a threshold for the p-value (a higher alpha allows more uncertainty in concluding that the null-hypothesis should be rejected). Because the t-test show a significant p-value for alpha=0.05 (p<alpha) we can safely reject the null-hypothesis and accept the claim of the expert.

Alpha is involved in computing a confidence interval for the true mean of the birthweights. The outcome of the t-test shows a confidence interval of 95% (alpha=0.05) that the true population mean lies in the range [2829.202, infinity]. For a confidence interval of 90% (alpha=0.10) the value for the lower bound of the confidence interval becomes a bit higher. This makes sense since the higher alpha, the more uncertainty there is about the true mean lying in the interval, thus the interval can become narrower (lower bound increases).

### c

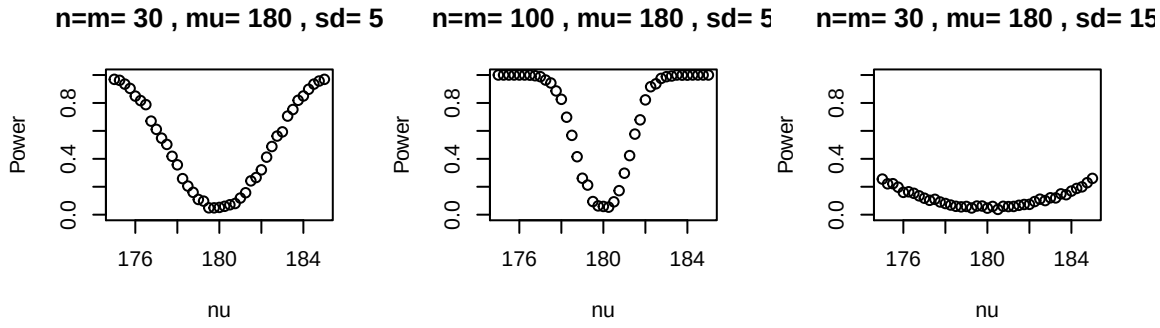
The 90% confidence intervals differ with respect to questions **a** and **b** since now a one-sided t-test is performed. This results in two differences:

- The total alpha in both questions is 0.1. But in **b** this alpha is only on one side of the t-distribution because it's a one sided test (giving one area in the tail of alpha=0.1) while in question **a** it is divided over both sides since it is a two sided test (giving two areas in the tails of both alpha=0.05).
- The one sided version means we only compute whether a distribution lies above (or below) a value, but not both at the same time. So basically we only asked whether the value is higher than the value in H0. Therefore the upper bound is not computed because it's irrelevant for the hypothesis that is tested. Hence the upper bound is shown as infinity.

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

### a & b & c

```r
n=m=30; mu=180; nu=175; sd=5
## Return power function
returnpower <- function(n,m,mu,nu,sd,B=1000){
  p=numeric(B) # p will be an array of realized p-values
  for (b in 1:B) {x=rnorm(n,mu,sd); y=rnorm(m,nu,sd)
                  p[b]=t.test(x,y,var.equal=TRUE)[[3]]}
  powerp=mean(p<0.05)
  return(powerp)}
## Define the lists for the sequence of nu
nuseq = seq(175,185,by=0.25)
## Calculate the power for multiple nu
plotpower <- function(n,m,mu,nuseq,sd){
  powernu = numeric(length(nuseq))
  for(a in 1:length(nuseq)){
    powernu[a] = returnpower(n,m,mu,nuseq[a],sd)
  }
  {plot(nuseq,powernu, main=paste("n=m=", n,", mu=",mu,", sd=", sd  ),
        xlab = "nu", ylab="Power", ylim=c(0,1))}
}
## Plot
par(mfrow=c(1,3))
plotpower(n,m,mu,nuseq,sd)
n=m=100; mu=180 ; sd=5
plotpower(n,m,mu,nuseq,sd)
n=m=30; mu=180 ; sd=15
plotpower(n,m,mu,nuseq,sd)
```




## d

From the above graphs we can witness the influence of the parameters on the power. Since mu=180 and we are measuring the power of nu ideally we would want a power graph that goes straight down from 1.0 at just before 180, and straight up from 0 to 1.0 just after 180. That would mean it would binary discriminate whether mu is equal to nu.

The difference between the situations plotted in a and b is the amount of sampled observations. In 'a' 30 observations are used for determining with a t-test whether the two distributions are equal, in 'b' 100 values are observed. From the figures it becomes clear that the power for the t-test is often higher in plot b than in plot a (when the distributions are actually different). This means that more hypotheses are correctly rejected when more data is available. This makes sense since the more samples there are, the surer we can be that two distributions are not from the same population (when in fact they are not).

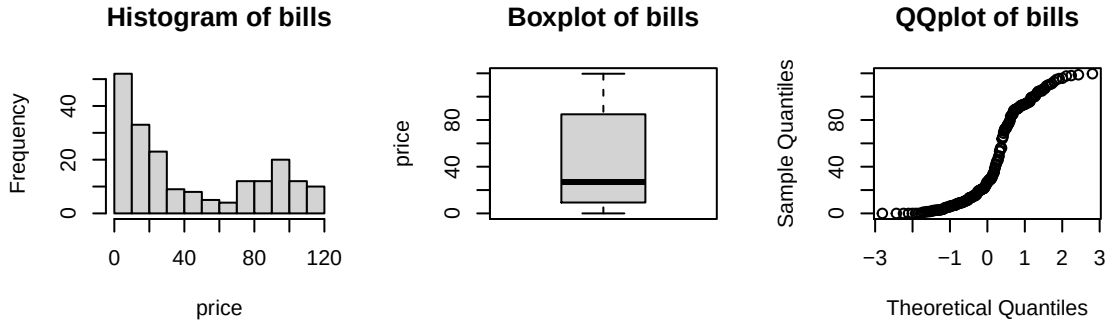
The difference between a and c is that the standard deviation is much higher in c (sd = 15) than in a (sd = 5). This causes the data to vary more broadly around the mean. This makes it harder to conclude that two distributions are different, which can be observed in the plots for a and c. Because of the high standard deviation in c the power does not get much higher than 0.25 (meaning 1 out of 4 hypotheses that should be rejected is rejected), while in a the power reaches values of almost one (meaning all hypotheses that should be rejected are rejected).

# Exercise 3. Telecommunication company

```
data = read.table("telephone.txt", header=TRUE)
```

### a

```
plot3(data$Bills, "bills", 'price')
```




From the boxplot, histogram and qqplot we can see the data is not normally distributed. Most of the bills include payments of 0-10 euros, the more expensive the bill becomes the less frequent it occurs. However bills become more frequent around 90-100 euros. We would advise this company to offer more products with a cost price in between the two peaks of the histogram. Now customers might be under the impression that the mobile phones of this shop are either very cheap or very expensive.

There exist a few 0 values in the data set, meaning that nothing is payed during a transaction. As long as the company is not really selling free mobile phones, this seems to be odd data. This influences the statistics of the data whithout providing extra information. Therefore this data is taken out of the dataset.

```
no_zeroes_data = data$Bill[data$Bill>0]
```

### b

```
bootstrap <- function(t){
  B=1000
  tstar=numeric(B)
  n=length(no_zeroes_data)
  lambdas = seq(0.01,0.1,0.01)
```

```
  pl=pr=p=numeric(length(lambdas))
  for(lambda in lambdas){
    for(i in 1:B){
      xstar=rexp(n,lambda)
      tstar[i]=median(xstar)
      }
    pl[lambda*100]=sum(tstar<t)/B
    pr[lambda*100]=sum(tstar>t)/B
    p[lambda*100]=2*min(pl[lambda*100],pr[lambda*100])
  }
  p
}

bootstrap( median(no_zeroes_data))
```

```
##  [1] 0.000 0.092 0.022 0.000 0.000 0.000 0.000 0.000 0.000 0.000
```

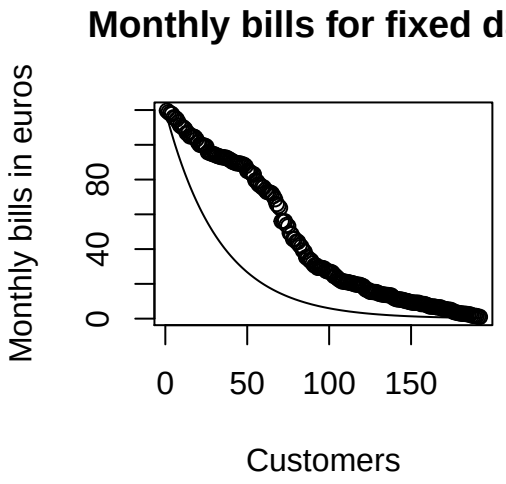
```
bootstrap( median(data$Bills))
```

```
##  [1] 0.000 0.028 0.122 0.000 0.000 0.000 0.000 0.000 0.000 0.000
```

From the above data we can conclude that if we take the median from the 'no-zeroes-data' there is significant result that the data stems from the population of exp(0.03). If we take the median from the data of all bills there is significant result that the data stems from exp(0.02). If we plot this we indeed see the samples somewhat resembling the exponential distributions. Note that we scale the exponential here to allow for the easiest to read graph. Where the exponential function usually intersects the y-axis at 1 we now scale it such that it intersects the y-axis at the same place as the telephone function:

```
par(mfrow=c(1,2))
no_zeroes_data <- sort(no_zeroes_data, decreasing = TRUE)
{plot(seq(1,length(no_zeroes_data),by=1), no_zeroes_data, xlab = "Customers",
      ylab = "Monthly bills in euros", main = "Monthly bills for fixed data")}
lines(max(no_zeroes_data)*dexp(0.03*seq(1,length(no_zeroes_data),by=1)))

phonebills <- sort(data$Bills, decreasing = TRUE)
{plot(seq(1,length(phonebills),by=1), phonebills, xlab = "Customers",
      ylab = "Monthly bills in euros", main = "Monthly bills for all data")}
lines(max(phonebills)*dexp(0.02*seq(1,length(phonebills),by=1)))
```





**c**

```
CI95median <- function(x){
    B=1000
    tstar = numeric(B)
    t1=median(x)
    for (i in 1:B){
      xstar=sample(x, replace=TRUE)
      tstar[i]=median(xstar)}
    tstar50=quantile(tstar,0.05)
    tstar950=quantile(tstar,0.95)
    ci=c((2*t1)-tstar950, (2*t1)-tstar50)
    ci
}
t1 = median(no_zeroes_data)
ci = CI95median(no_zeroes_data)
```

The 95% bootstrap confidence interval for the population median of bills is [19.341, 36.145] around its median: 28.905 .

**d**

```
estimated_mean = mean(no_zeroes_data) #sample mean
estimated_mean
```

```
## [1] 45.4
```

We can compute an estimate of the population mean according to the Central Limit Theorem for the sample mean. This estimated mean can be used in order to find the value for lambda in the exponential distribution since E[mean]=1/lambda. Thus lambda=1/E[mean].

```r
# based on sample mean we can compute lambda
lambda = 1/estimated_mean
lambda
```

```
## [1] 0.022
```

```r
# computing the CI using the above formula:
n=length(no_zeroes_data)
mediaan = log(2)/lambda
std = sqrt(1/lambda**2)
ci = qt(0.95,df=n-1)*std/sqrt(n)
upperbound = mediaan + ci
lowerbound = mediaan - ci
upperbound;lowerbound;mediaan
```

```
## [1] 36.9
```

```
## [1] 26.1
```

```
## [1] 31.5
```

So the 95% confidence interval of the population median is [36.887, 36.887] around its median 31.471.

**e**

```r
more = no_zeroes_data>=40
more2 = sum(more, na.rm = TRUE)
binom.test(more2,length(no_zeroes_data))
```

```
##
##  Exact binomial test
##
## data:  more2 and length(no_zeroes_data)
## number of successes = 83, number of trials = 192, p-value = 0.07
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.361 0.506
## sample estimates:
## probability of success
##                  0.432
```

```r
ten_test = no_zeroes_data<10
ten = sum(ten_test, na.rm=TRUE)
binom.test(ten,length(no_zeroes_data))
```

```
##
##  Exact binomial test
##
## data:  ten and length(no_zeroes_data)
```

```
## number of successes = 44, number of trials = 192, p-value = 2e-14
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.172 0.295
## sample estimates:
## probability of success
##                  0.229
```

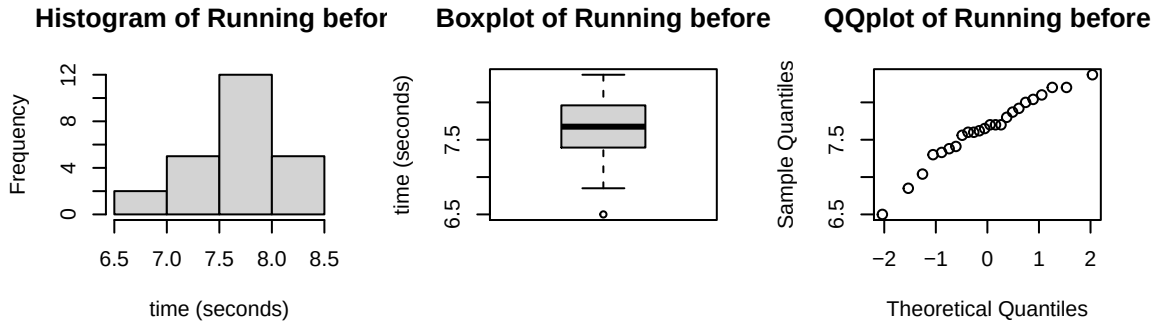
We have used the sign test for this problem, because we're working with one sample which is not normally distributed. From the test, we can conclude that the null hypothesis of the median being greater or equal to 40 should not be rejected, because the p-value is 0.07, which is greater than the alpha of 0.05.

In the second sign test, it is evident that the probability of success is 22.9% of the bill being less than 10 euro, which is of course less than the proposed 25%.

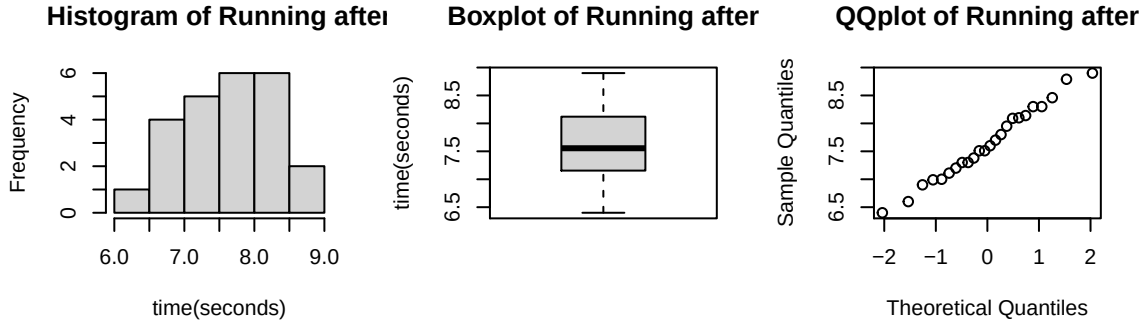
# Exercise 4: Energy drink

## a

```
run <- read.table("run.txt", header=TRUE)
plot3(run$before, "Running before", 'time (seconds)')
```




```
plot3(run$after, "Running after", 'time(seconds)')
```



```r
shapiro.test(run$after);shapiro.test(run$before)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  run$after
## W = 1, p-value = 0.9

##
##  Shapiro-Wilk normality test
##
## data:  run$before
## W = 1, p-value = 0.4
```

```r
cor.test(run$before, run$after, method="spearman")
```

```
## Warning in cor.test.default(run$before, run$after, method = "spearman"): Cannot
## compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  run$before and run$after
## S = 859, p-value = 0.001
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##   rho
## 0.626
```

There is significant correlation. When looking at the qqplot of 'before' the normality looks doubtful, whereas the qqplot of 'after' looks more normal. Both boxplots look quite normal. After's histogram looks a bit more doubtful but there are very little data points. Because of the qqplot we use the Spearman's rank correlation test (even though the Shapiro-Wilk test shows a p-value above 0.05, indicating the assumption of the normal distribution cannot be rejected, it seemed more appropriate to use the spearman rank test).

## b

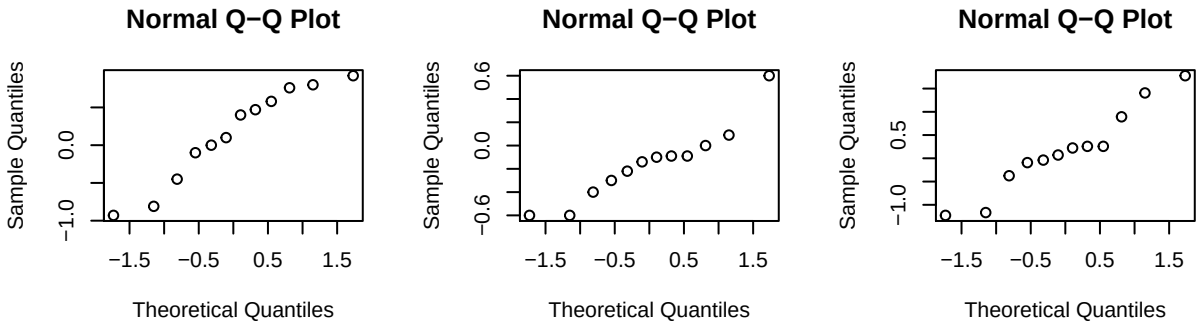
```r
library(dplyr)
```

```
##
## 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
```

```
par(mfrow=c(1,3))
lemonade <- run %>%
  filter(drink=="lemo")
qqnorm(lemonade$after-lemonade$before)
energy <- run %>%
  filter(drink=="energy")
qqnorm(energy$after-energy$before)
#check is a previously sampled simulated normal distribution
{check=c(-0.03844319513, -0.09277258091, 1.77619809815, 0.22196557385, 1.40569961661,
         -1.16692748524, -1.22750103059, 0.89233707536, -0.37489419067, 0.07086510670,
         0.25706338223, 0.25732333745)}
qqnorm(check)
```




```
t.test(lemonade$before, lemonade$after, paired=TRUE)[[3]]
```

```
## [1] 0.437
```

```
t.test(energy$before, energy$after, paired=TRUE)[[3]]
```

```
## [1] 0.126
```

For this problem, we are using the t-test, because the qqplots show that for both drinks, the difference in running times are normally distributed. While the qqplot for energy is doubtful, when sampling only 12 data points from a simulated normal distribution with the same degrees of freedom, one of the outcomes is the third graph, which looks a lot like the qqplot for energy. There seems to be a difference in running speeds, which is not significant however, since both p-values are above 0.05. The null-hypothesis (true difference in means is equal to zero) is thus not rejected. It is interesting to see that the energy group has a positive mean of the differences, indicating a decrease in running speed, whereas the lemonade group has an increase in running speed.

**c**

```
par(mfrow=c(1,2))
run$difference=(run$after-run$before)
lemo_diff <- run[which(run$drink == 'lemo'),]
ener_diff <- run[which(run$drink == 'energy'),]
t.test(lemo_diff$difference,ener_diff$difference)[[3]]
```

```
## [1] 0.159
```

Because the difference in running times is normally distributed, again the two sample t-test is used. The value that is printed is the p-value, which is higher than 0.05, so we can't reject the null hypothesis (which is that the means of the two populations are the same).

### d

If the main aim was to test whether drinking energy drink speeds up running you could wonder why the control group drank lemonade. A plausible reason for increased running speed from drinking energy could be the sugar that it is notorious for, but there is also sugar in lemonade. A better control criteria would to just let the control group drink water to just allow them to hydrate.

In (a) we found that the values are correlated, which makes sense since they are are measured on the same persons. in (b) we found different speeds for both groups but not one of them turned out to be significantly relevant.

In (c) we rewrote the two-sample t-test to a paired t-test and obtained the same results.

## Exercise 5: Chickweights

### a

The data is unpaired since we have two groups of experimental unit. It would be the same as testing different types of fertilizer on a plot of land (which would be subdivided for the two groups). Also the samples are of different size.

```
chickweights <- chickwts
meatmeal <- chickweights %>% filter(feed=="meatmeal")
sunflower <- chickweights %>% filter(feed=="sunflower")
t.test(meatmeal$weight, sunflower$weight)[[3]]
```

```
## [1] 0.0444
```

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

```
##
##  Wilcoxon rank sum exact test
##
## data:  meatmeal$weight and sunflower$weight
## W = 36, p-value = 0.07
## alternative hypothesis: true location shift is not equal to 0
```

```
ks.test(meatmeal$weight, sunflower$weight)[[2]]
```

```
## [1] 0.108
```

The outcome of the unpaired t test is significant with a p-value of 0.04. The null-hypothesis could thus be rejected, meaning that there is a true difference in means not equal to 0. From the results we see that the mean chickenweight for sunflower is bigger. The Welch two-sample t-test assumes both samples come

from a normal population however. Given the small sample size one could argue this for meatmeal, but the normality of sunflower seems very doubtful.

For the Mann-Whitney test, the outcome of the test is insignificant, with a p-value of 0.07. This means that the null-hypothesis is not rejected and the difference in medians is 0.

The Kolmogorov-Smirnov test also yields an insignificant result, with a p-value of 0.1. Now the H-0 of equal means is not rejected, meaning the means of the two populations are similar.

So as we use more powerful tests and more specific for a-normal data, the p-values increase. Only the unpaired t-test is significant but this test relies on normality, which the data is not. From the Mann-Whitney test especially we know the medians are equal, and from the Kolmogorov-Smirnov test we know the means are equal. The last test is the most powerful, especially for this a-normal data.

## b

```
is.numeric(chickweights$weight); is.factor(chickweights$feed)
```

```
## [1] TRUE
```

```
## [1] TRUE
```

```
chickweights_aov=lm(weight~feed, data=chickweights)
anova(chickweights_aov)
```

```
## Analysis of Variance Table
##
## Response: weight
##           Df Sum Sq Mean Sq F value  Pr(>F)
## feed       5 231129   46226    15.4 5.9e-10 ***
## Residuals 65 195556    3009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As can be seen from the small p value (5.9e-10) the Null Hypothesis is rejected, meaning that the different types of supplement do have a significant effect on the weight of the chicks.

```
summary(chickweights_aov)
```

```
##
## Call:
## lm(formula = weight ~ feed, data = chickweights)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -123.91  -34.41    1.57   38.17  103.09
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     323.58      15.83   20.44  < 2e-16 ***
## feedhorsebean  -163.38      23.49   -6.96  2.1e-09 ***
```

```
## feedlinseed    -104.83      22.39   -4.68  1.5e-05 ***
## feedmeatmeal    -46.67      22.90   -2.04  0.04557 *
## feedsoybean     -77.15      21.58   -3.58  0.00067 ***
## feedsunflower     5.33      22.39    0.24  0.81249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.9 on 65 degrees of freedom
## Multiple R-squared:  0.542,  Adjusted R-squared:  0.506
## F-statistic: 15.4 on 5 and 65 DF,  p-value: 5.94e-10
```

The estimated chick weights for each of the supplements are:

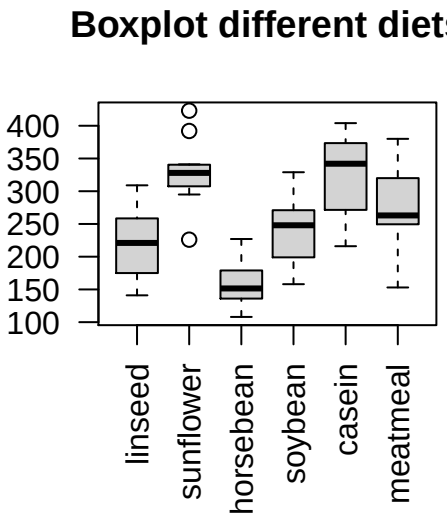
| Supplement | Casein | Horsebean | Linseed | Meatmeal | Soybean | Sunflower |
|---|---|---|---|---|---|---|
| weight | 323.58 | 160.2 | 218.8 | 276.9 | 246.4 | 328.9 |

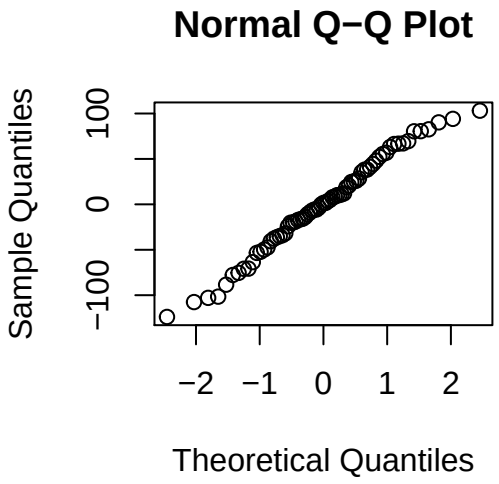
The best supplement seems to be sunflower with an estimated mean of 328.9 grams. The second best is casein with an estimated mean of 323.58, with an insignificant difference between them. Statistically seen these two supplements thus share the first place. From the above summary we see that all other supplements do differ significantly and are all way lower than the best two.

**c**

The assumptions in the one-way Anova are that samples are obtained independently from normal populations with equal variances.

```
par(mfrow=c(1,2))
# Check variance
boxplot(chickweights$weight[chickweights$feed=='linseed'],
        chickweights$weight[chickweights$feed=='sunflower'],
        chickweights$weight[chickweights$feed=='horsebean'],
        chickweights$weight[chickweights$feed=='soybean'],
        chickweights$weight[chickweights$feed=='casein'],
        chickweights$weight[chickweights$feed=='meatmeal'],
        main="Boxplot different diets", las = 2,
        names = c('linseed', 'sunflower', 'horsebean', 'soybean', 'casein', 'meatmeal'))
# Check normality
qqnorm(residuals(chickweights_aov))
```




```
shapiro.test(residuals(chickweights_aov))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(chickweights_aov)
## W = 1, p-value = 0.6
```

The first assumption to be investigated is that the variance of the weights in the different diet groups should be approximately the same. From the boxplot we can see that this is true (box sizes and whiskers don't differ too much) except for the 'sunflower' diet.

According to the QQ-plot for the residuals of all the different diets (corrected for being sampled from different populations) the data seems to be approximately normally distributed, since most datapoints fall approximately around the diagonal. The other assumption is that the variance of the data in each diet is approximately the same. Additionally the results of the shapiro test do not provide evidence for non-normally distributed weights.

### d

```
attach(chickweights); kruskal.test(weight, feed)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  weight and feed
## Kruskal-Wallis chi-squared = 37, df = 5, p-value = 5e-07
```

Kruskal-Wallis is a generalisation of Mann-Whitney but over multiple treatments. The only assumption is that for each group (diet) there are more than 5 measurements available (this is the case). The above performed Kruskal-Wallis test shows that also for the non-parametric version of the Anova the F-statistics of some different diets is not equal. This is a less strong outcome than the one obtained by the Anova test, since we do not know which groups are significantly different (as oposed to the Anova test).