Assignment1

```
set.seed(1234)
```

Assignment 1

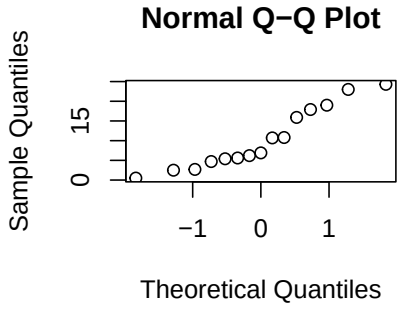
Jop Keuning 11014407, Nienke Deutz 2732866 and Harshita Choudhary 13807609

Exercise 1

1a

First of we check the data for normality which is done in two ways, first with a qqplot which shows that normality is present.

```
waiting_times <- c(5.4, 17.9, 19.0, 0.5, 15.9, 2.7, 6.2, 2.5, 4.7, 6.9, 10.8, 24.3, 5.6, 23.0, 10.7)
mu <- mean(waiting_times)
sigma <- sd(waiting_times)
qqnorm(waiting_times)
```



But to be extra sure that normality is indeed the case a shapiro test will confirm it as indeed fufilling the normality condition

```
shapiro.test(waiting_times)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  waiting_times
## W = 0.90744, p-value = 0.1237
```

Now that normality has been confirmed it is time to construct a confidence interval for the mu at 97% for which alpha needs to be set at 0.05. With this the confidence interval and the minimum sample size can be found.

To calculate the confidence interval we take the mean of the data we take upper quantile of the normal distribution as za with the upper quantile taken defined as 1-alpha. With this the confidence is calculated as $mean - (za/2) * sd/squareroot(n)$ as the lower bound and $mean + (za/2) * sd/squareroot(n)$ where n is the sample size set at 10, mean is the average of the data and sd the standard deviation

```
n <- length(waiting_times)

alpha <- 0.015
za <- qnorm(1-alpha)
ci <- c(mu -(za/2)*(sigma/sqrt(n)), mu + (za/2)*(sigma/sqrt(n)))
print(ci)
```

```
## [1]  8.23295 12.58038
```

To calculate the minimum sample size needed to attain this confidence interval a za value of the normal distribution of $1 - alpha/2$ is needed. With this the calculation of the minimum number of samples is as follows: $n_min = (za^2 * sd^2)/E$. Here E signifies the error and the the confidence interval will have a length of at most 2E. As we want the length to be at most 2 we can set E equal to 1 resulting in the followint minimum number of samples

```
za2 <- qnorm(1-alpha/2)
n_minimum <- (za2^2*sigma^2)/1
print(n_minimum)
```

```
## [1] 356.1753
```

Finally we want to compare the previously calculated confidence interval with a bootstrapped one. The code for this bootstrap and it's result can be seen below.

```
B = 1000
Tstar = numeric(B)
for(i in 1:B){
  Xstar <- sample(waiting_times, replace = TRUE)
  Tstar[i] <- mean(Xstar)}
Tstar015 <- quantile(Tstar, 0.015)
Tstar985 <- quantile(Tstar, 0.985)
sum(Tstar <Tstar015)
```

```
## [1] 15
```

```
boodstrap <- c(2*mu-Tstar985,2*mu-Tstar015)
print(boodstrap)
```

```
##     98.5%      1.5%
##  6.252133 14.353533
```

When compared to the confidence interval calculated above which looks like this, (8.23295 12.58038) we can see that the bootstrap confidence interval is wider than that of the calculated one by a difference of around 2 in both bounds.

1b.

To verify the doctor's claim we perform both a t-test and a sign test for a mean less than 15. From the t-test we can see that the mean lies below 13.94 with a confidence interval of 95%. With the sign test we get a probability range of 0.69 to 1 with the confidence interval of 95% meaning we can say that for both tests the doctor's claim that the waiting time is less than 15 minutes on average is correct.

```
ttest = t.test(waiting_times,mu=15,alt="less")
print(ttest)
```

```
##
##  One Sample t-test
##
## data:  waiting_times
```

```
## t = -2.2928, df = 14, p-value = 0.01893
## alternative hypothesis: true mean is less than 15
## 95 percent confidence interval:
##      -Inf 13.93517
## sample estimates:
## mean of x
##  10.40667
```

```
n = 10
binom.test(sum(waiting_times<15), n)
```

```
##
##  Exact binomial test
##
## data:  sum(waiting_times < 15) and n
## number of successes = 10, number of trials = 10, p-value = 0.001953
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6915029 1.0000000
## sample estimates:
## probability of success
##                      1
```

1c.

To compute the power of the t-test and sign test is to use p-value of different samples from the data for a large amount of iterations and calculate the average of all p-values that satisfy the condition of $p_value < 0.05$, meaning all significant p-values. This results in that the sign tests have a power of 1 but in both the cases for mu at 13 and 14 the ttest has a power below 0.5

```
B=1000
n<-length(waiting_times)
psign<-numeric(B)
pttest<-numeric(B)
for(i in 1:B) {
  x = sample(waiting_times, replace = TRUE)
  pttest[i]<-t.test(x, mu=13)[[3]] ## extract p-value
  psign[i]<-binom.test(sum(x=13), n, p=0.5)[[3]]
}
sum(psign<0.05)/B
```

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

```
sum(pttest<0.05)/B
```

```
## [1] 0.264
```

```
for(i in 1:B) {
  x = sample(waiting_times, replace = TRUE)
  pttest[i]<-t.test(x, mu=14)[[3]] ## extract p-value
  psign[i]<-binom.test(sum(x=14), n, p=0.5)[[3]]
}
sum(psign<0.05)/B
```

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

```
sum(pttest<0.05)/B
```

```
## [1] 0.407
```

1d.

$CL = \hat{p} \mp z_{a/2} \cdot 2\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$
$n = 15$
Values > 15.5 : $17.9, 19.0, 15.9, 24.3, 23.0$
Thus: $\hat{p} = \frac{5}{15} = \frac{1}{3}$

$\hat{p}_r = 0.53$, therefore:
$\hat{p} + z_{a/2} \cdot 2\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} = 0.53$

$\frac{1}{3} + z_{a/2} \cdot 2\sqrt{\frac{\frac{1}{3}(1-\frac{1}{3})}{15}} = 0.53$

$\frac{1}{3} + z_{a/2} \cdot 2\sqrt{\frac{2}{135}} = 0.53$

$z_{a/2} \cdot 2\sqrt{(\frac{2}{135})} = 0.20$

$z_{a/2} = 0.20/2\sqrt{\frac{2}{135}}$

$z_{a/2} = 0.82$

This means that:

$\hat{p}_l = \frac{1}{3} - 0.82 \cdot 2\sqrt{\frac{2}{135}} = 0.13$

A $z_{a/2} = 0.82$ corresponds to a Confidence level of 79%

1e.

We create two new populations from the data, one for women and one for men. Then, we convert the data to binary: 0 for people with waiting time greater than 15.5 minutes, 1 otherwise. The populations are then: Women: [0,0,1,1,1,1,1,1] Men: [0,0,0,1,1,1,1]

The best way to compare these independent populations is with an **independent sample t-test**. Here, H_0 = The waiting times for men and for women are equal, and H_1 = The waiting times for men and for women differ. Running this test in R, we get:

```
men <- c(0,0,0,1,1,1,1)
women <- c(0,0,1,1,1,1,1,1)
t.test(men, women)
```

```
##
##  Welch Two Sample t-test
##
## data:  men and women
## t = -0.6868, df = 12.021, p-value = 0.5052
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.7449595  0.3878166
## sample estimates:
## mean of x mean of y
## 0.5714286 0.7500000
```

The p-value is larger than 0.05, and therefore we cannot reject the null hypothesis. Thus, the researcher is incorrect when they state that the waiting times for men and women are different.

Exercise 2

2a.

First of all, the data is independent and not paired. This is because no data point in the unseeded population has a correlations with a data point in the seeded population. For each test, we define: H_0: the addition of silver nitrate does not lead to more rainfall, and H_1: the addition of silver nitrate leads to more rainfall. We perform the three tests:

```
clouds <- read.csv("clouds.txt", sep="")
t.test(clouds[1:26,1], clouds[1:26,2])
```

```
##
##  Welch Two Sample t-test
##
## data:  clouds[1:26, 1] and clouds[1:26, 2]
## t = 1.9984, df = 33.856, p-value = 0.05375
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   -4.740491 559.585876
## sample estimates:
## mean of x mean of y
##  441.9846  164.5619
```

```
wilcox.test(clouds[1:26,1], clouds[1:26,2])
```

```
## Warning in wilcox.test.default(clouds[1:26, 1], clouds[1:26, 2]): cannot compute
## exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  clouds[1:26, 1] and clouds[1:26, 2]
## W = 473, p-value = 0.01383
## alternative hypothesis: true location shift is not equal to 0
```

```
ks.test(clouds[1:26,1], clouds[1:26,2])
```

```
## Warning in ks.test(clouds[1:26, 1], clouds[1:26, 2]): cannot compute exact p-
## value with ties
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  clouds[1:26, 1] and clouds[1:26, 2]
## D = 0.42308, p-value = 0.01905
## alternative hypothesis: two-sided
```

Looking at the results, we can reject the null hypothesis according to the Mann-Whitney and Kolmogorov-Smirnov tests, as their p-values are less than 0.05. However, according to the t-test, we cannot reject the null hypothesis, as its p-value is greater than 0.05. In this case, both the Mann-Whitney and Kolmogorov-Smirnov tests are not suitable for the task, since they assume the samples are taken from different populations. However, the samples are taken from the same population of clouds. Therefore, we consider the t-test correct, which states that we cannot reject the null hypothesis, and therefore we cannot state that the silver nitrate has an effect.

2b.

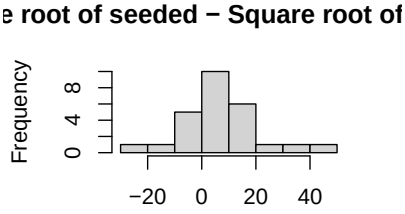
```
clouds <- read.csv("clouds.txt", sep="")
is.data.frame(clouds)
```

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

```
clouds_copy <- data.frame(clouds)
clouds_copy$square_root_seeded ='^'(clouds_copy$seeded,1/2)
clouds_copy$square_root_unseeded ='^'(clouds_copy$unseeded,1/2)
clouds_copy$double_square_root_seeded='^'(clouds_copy$square_root_seeded,1/2)
clouds_copy$double_square_root_unseeded='^'(clouds_copy$square_root_unseeded,1/2)
#View(clouds_copy)
```

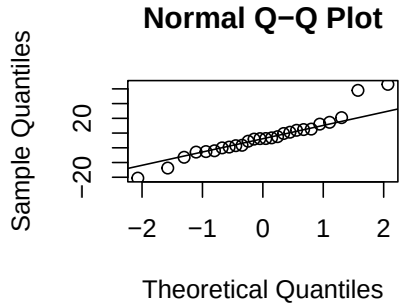
Square root of the values:

```
# FOR NORMALITY CHECK
hist(clouds_copy[['square_root_seeded']] - clouds_copy[['square_root_unseeded']],main="Square root of s
```




```
uare_root_seeded"]] − clouds_copy[["squ
```

```
# FOR NORMALITY CHECK
qqnorm(clouds_copy[['square_root_seeded']]-clouds_copy[['square_root_unseeded']]); qqline(clouds_copy[[
```



QQ - data deviates at the tail. It doesn't seem to follow a normal distribution. There are a few outliers.

Two Sample T-test:

```
# T-test for two paired samples - root
t.test(clouds_copy[['square_root_seeded']],clouds_copy[['square_root_unseeded']],paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  clouds_copy[["square_root_seeded"]] and clouds_copy[["square_root_unseeded"]]
## t = 2.682, df = 25, p-value = 0.01278
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   1.656326 12.617061
## sample estimates:
## mean of the differences
```

```
##                 7.136693
```

p-value is 0.0127 therefore, H0 is rejected. Hence, nitrate has an effect.

Mann-Whitney Test:

```
## Mann-Whitnney Test
wilcox.test(clouds_copy[['square_root_seeded']],clouds_copy[['square_root_unseeded']])
```

```
## Warning in wilcox.test.default(clouds_copy[["square_root_seeded"]],
## clouds_copy[["square_root_unseeded"]]): cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  clouds_copy[["square_root_seeded"]] and clouds_copy[["square_root_unseeded"]]
## W = 473, p-value = 0.01383
## alternative hypothesis: true location shift is not equal to 0
```

p-value is 0.0138 less than 0.05 therefore, H0 is rejected. Hence, nitrate has an effect.

Kolmogorov-Smirnov Test:

```
## kolmogorov - smirnov Test
ks.test(clouds_copy[['square_root_seeded']],clouds_copy[['square_root_unseeded']])
```

```
## Warning in ks.test(clouds_copy[["square_root_seeded"]],
## clouds_copy[["square_root_unseeded"]]): cannot compute exact p-value with ties
```

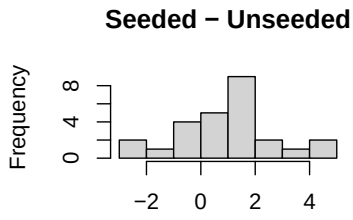
```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  clouds_copy[["square_root_seeded"]] and clouds_copy[["square_root_unseeded"]]
## D = 0.42308, p-value = 0.01905
## alternative hypothesis: two-sided
```

```
# p-value is less than 0.05 therefore, H0  rejected
```

p-value is 0.01905 less than 0.05 therefore, H0 is rejected. Hence, nitrate has an effect.

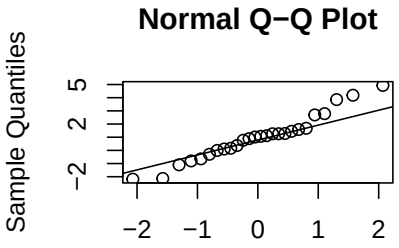
Double square root values:

```
# FOR NORMALITY CHECK
hist(clouds_copy[['double_square_root_seeded']]-clouds_copy[['double_square_root_unseeded']],main='Seed
```



uare_root_seeded"]] − clouds_copy[["dou

```
# FOR NORMALITY CHECK
qqnorm(clouds_copy[['double_square_root_seeded']]-clouds_copy[['double_square_root_unseeded']])
qqline(clouds_copy[['double_square_root_seeded']]-clouds_copy[['double_square_root_unseeded']])
```



QQ : Data deviates at the tails so distribution doesn't seem to be normal.

Two Sample T-Test:

```
# T-test for two paired samples - double root
t.test(clouds_copy[['double_square_root_seeded']],clouds_copy[['double_square_root_unseeded']],paired=TR
```

```
##
##  Paired t-test
##
## data:  clouds_copy[["double_square_root_seeded"]] and clouds_copy[["double_square_root_unseeded"]]
## t = 2.8413, df = 25, p-value = 0.008811
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2673422 1.6759523
## sample estimates:
## mean of the differences
##               0.9716472
```

p-value is 0.008811 therefore, H0 : no effect of silver nitrate, can be rejected. Hence, nitrate has an effect.

Mann-Whitnney Test:

```
## Mann-Whitnney Test
wilcox.test(clouds_copy[['double_square_root_seeded']],clouds_copy[['double_square_root_unseeded']])
```

```
## Warning in wilcox.test.default(clouds_copy[["double_square_root_seeded"]], :
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  clouds_copy[["double_square_root_seeded"]] and clouds_copy[["double_square_root_unseeded"]]
## W = 473, p-value = 0.01383
## alternative hypothesis: true location shift is not equal to 0
```

p-value is 0.01383. Hence, silver nitrate has an effect.

Kolmogorov - Smirnov Test:

```
## kolmogorov - smirnov Test
ks.test(clouds_copy[['double_square_root_seeded']],clouds_copy[['double_square_root_unseeded']])
```

```
## Warning in ks.test(clouds_copy[["double_square_root_seeded"]],
## clouds_copy[["double_square_root_unseeded"]]): cannot compute exact p-value with
## ties
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  clouds_copy[["double_square_root_seeded"]] and clouds_copy[["double_square_root_unseeded"]]
```

```
## D = 0.42308, p-value = 0.01905
## alternative hypothesis: two-sided
```

p-value is 0.01905. Hence, silver nitrate has an effect.

Conclusion: The p-value of two sample T-test is lower when the square root of the data is taken. However, for Mann Whitney and Kolmogorov - Smirnov Test, the p-value doesn't change on taking the square root.

2c.

```
#Calculate the  mean
mean_estimator = mean(clouds$seeded)
std_dev_estimator = sd(clouds$seeded)
n = 26
sqrt(n)
```

```
## [1] 5.09902
```

```
t_0.025 = 2.05954
c_i_lower = mean_estimator - t_0.025*std_dev_estimator/sqrt(n)
c_i_upper = mean_estimator + t_0.025*std_dev_estimator/sqrt(n)
c_i_lower
```

```
## [1] 179.1258
```

```
c_i_upper
```

```
## [1] 704.8434
```

```
t.test(clouds[['seeded']],mu = 0)
```

```
##
##  One Sample t-test
##
## data:  clouds[["seeded"]]
## t = 3.463, df = 25, p-value = 0.001937
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  179.1260 704.8432
## sample estimates:
## mean of x
##  441.9846
```

```
estimator_lambda = 1/mean_estimator
ci_lower = 1/704.8432
ci_upper = 1/179.1260
estimator_lambda
```

```
## [1] 0.002262522
```

```
ci_lower
```

```
## [1] 0.001418755
```

```
ci_upper
```

```
## [1] 0.005582662
```

lambda(Estimator) = 0.002262522 and confidence interval is (0.001418755,0.005582662)

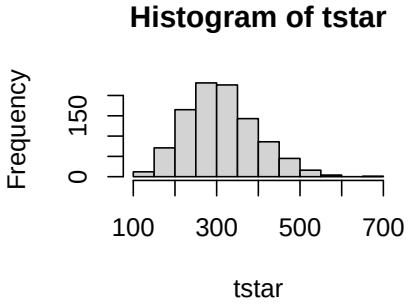
Bootstrap Test:

```
T = median(clouds[['seeded']])
T
```

```
## [1] 221.6
```

```
B=1000
tstar=numeric(B)
n=26

for (i in 1:B){
  xstar=rexp(n,estimator_lambda)
  tstar[i]=median(xstar)}
hist(tstar)
```




```
pl=sum(tstar<T)/B; pr=sum(tstar>T)/B; p=2*min(pl,pr)
pl;pr;p
```

```
## [1] 0.144
```

```
## [1] 0.856
```

```
## [1] 0.288
```

We get p = 0.288 which is greater than 0.05 therefore, we can't reject the null hypothesis : seeded data belongs to exponential distribution of a rate parameter lambda.

kolmogorov - Smirnov Test:

```
## kolmogorov - smirnov Test
ks.test(x = clouds[['seeded']], y = "pexp", rate = estimator_lambda)
```

```
## Warning in ks.test(x = clouds[["seeded"]], y = "pexp", rate = estimator_lambda):
## ties should not be present for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  clouds[["seeded"]]
## D = 0.20035, p-value = 0.2476
## alternative hypothesis: two-sided
```

We get p = 0.2476 which is greater than 0.05 therefore, we can't reject the null hypothesis : that seeded data belongs to exponential distribution of a rate parameter lambda.

2d.

```
# Left tailed
# null hypothesis : median >= 300
# Alternate : median < 300
```

```
n = 26
x = sum(clouds[['seeded']] < 300 )
x
```

```
## [1] 17
```

```
binom.test(x,n,p=0.5)
```

```
##
##  Exact binomial test
##
## data:  x and n
## number of successes = 17, number of trials = 26, p-value = 0.1686
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.4433281 0.8278559
## sample estimates:
## probability of success
##              0.6538462
```

We get a p-value of 0.1686, therefore we can't reject the null hypothesis that the median is greater than or equal to 300. We can say with 95% significance level that the median is not less than 300.

```
num_less_30 = sum(clouds[['seeded']] < 30 )
n=26
binom.test(num_less_30,n,p=0.25,alternative="less")
```

```
##
##  Exact binomial test
##
## data:  num_less_30 and n
## number of successes = 3, number of trials = 26, p-value = 0.08019
## alternative hypothesis: true probability of success is less than 0.25
## 95 percent confidence interval:
##  0.000000 0.271902
## sample estimates:
## probability of success
##              0.1153846
```

p-value is 0.08019. Therefore, we reject that at most 25% of seeded clouds have precipitation less than 30 with significance level of 95%.

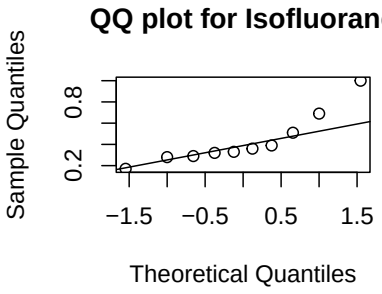
Exercise 3

```
dogs <- read.csv("dogs.txt", sep="")
is.data.frame(dogs)
```

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

3a.

```
# FOR NORMALITY CHECK
qqnorm(dogs[,1], main="QQ plot for Isofluorane"); qqline(dogs[,1])
```



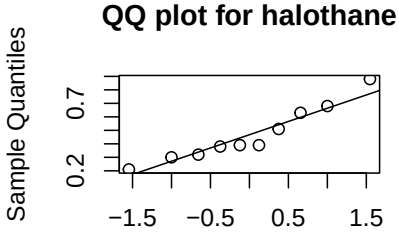
QQ plot deivaties at the tail. It doesn't seem to follow a normal distribution.

```
shapiro.test(dogs[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dogs[, 1]
## W = 0.83093, p-value = 0.03434
```

p-value of shapiro-test is less than 0.05 therefore the claim that it follows a normal distribution is rejected.

```
qqnorm(dogs[,2], main="QQ plot for halothane"); qqline(dogs[,2])
```



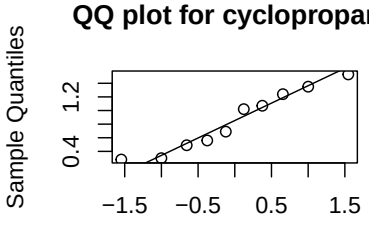
Looking at the QQ-plot it is reasonable to say that Halothane data is from a normal distribution.

```
shapiro.test(dogs[,2])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dogs[, 2]
## W = 0.9234, p-value = 0.3862
```

shapiro test p-value is greater than 0.05. It follows a normal distribution.

```
qqnorm(dogs[,3],main = "QQ plot for cyclopropane"); qqline(dogs[,3])
```



Very less diviation therefore, we can say that it comes from a normal distribution.

```
shapiro.test(dogs[,3])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dogs[, 3]
## W = 0.93334, p-value = 0.4815
```

shapiro test p-value is greater than 0.05. It follows a normal distribution.

3b.

CORRELATION TEST :

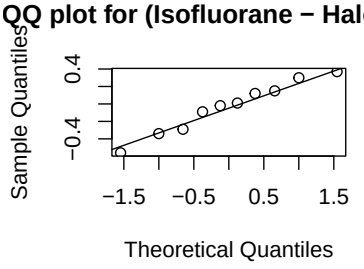
```
## correlation isofluorane and halothane
cor.test(dogs[['isofluorane']],dogs[['halothane']],method="spearman")
```

```
## Warning in cor.test.default(dogs[["isofluorane"]], dogs[["halothane"]], : Cannot
## compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  dogs[["isofluorane"]] and dogs[["halothane"]]
## S = 128.89, p-value = 0.5436
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.218846
```

p value of 0.5 is greater than 0.05 therefore, we can't reject the null hypothesis that there is no linear correlation. Hence, isofluorane and halothane are not linearly correlated.

```
qqnorm(dogs[,1]-dogs[,2] , main = "QQ plot for (Isofluorane - Halothane)")
qqline(dogs[,1]-dogs[,2])
```



QQ plot - Very less diviation therefore, we can say that it comes from a normal distribution.

```
shapiro.test(dogs[,1]-dogs[,2])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dogs[, 1] - dogs[, 2]
## W = 0.96828, p-value = 0.8744
```

shapiro test p-value is 0.8744 which is greater than 0.05. Therefore, the difference in isofluorane and halothane is normally distributed.

Two paired sample T-Test :

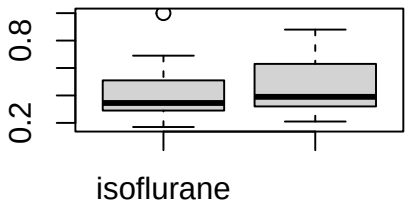
```
t.test(dogs[,1],dogs[,2],paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  dogs[, 1] and dogs[, 2]
## t = -0.37755, df = 9, p-value = 0.7145
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2447093  0.1747093
## sample estimates:
## mean of the differences
##                  -0.035
```

p value of 0.7145 is more than 0.05, therefore there is no difference between the effect of isofluorane and halothane. Distribution of these columns are not significantly different.

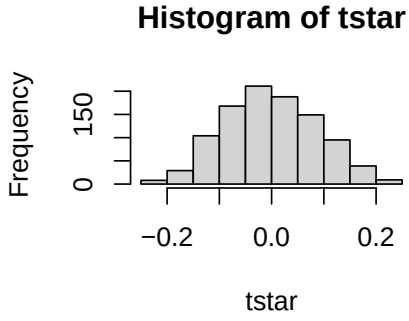
PERMUTATION TEST:

```
boxplot(dogs[,1],dogs[,2],names=c("isoflurane","halothane"))
```




```
mystat=function(x,y) {mean(x-y)}
B=1000; tstar=numeric(B)
for (i in 1:B) {
  dogstar=t(apply(cbind(dogs[,1],dogs[,2]),1,sample))
  tstar[i]=mystat(dogstar[,1],dogstar[,2]) }

myt=mystat(dogs[,1],dogs[,2])
#myt
hist(tstar)
```




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

```
## [1] 0.738
```

p value of 0.738 is more than 0.05, therefore there is no difference between the effect of isofluorane and halothane. Distribution of these columns are not significantly different

Permutation test for two-paired samples is valid here and it doesn't depend on the distribution of data. Here, T-test will perform better than the permutation test because data of difference in isofluorane and halothane is normally distributed.

3c.

One - way ANOVA:

```
dogsframe=data.frame(yield=as.vector(as.matrix(dogs)),variety=factor(rep(1:3,each=10)))

is.factor(dogsframe$variety); is.numeric(dogsframe$variety)
```

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

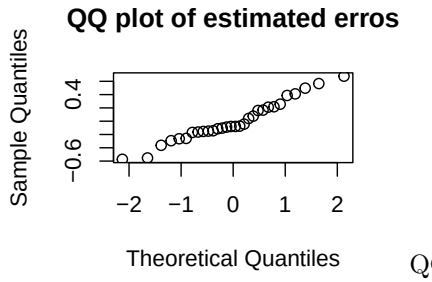
```
## [1] FALSE
```

```
dogsanova=lm(yield~variety,data=dogsframe)
anova(dogsanova)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value Pr(>F)
## variety    2 1.0808 0.54040   5.355  0.011 *
## Residuals 27 2.7247 0.10092
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

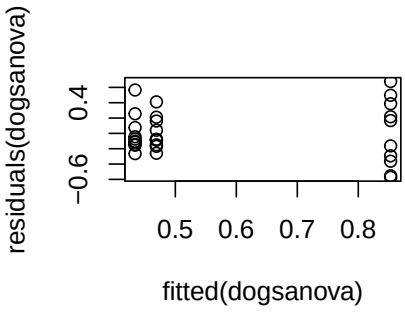
The p-value 0.011, hence HA : mu1 = mu2 = mu3 is is rejected with 95% significance level. Therefore, the type of drug has an effect on the concentration of plasma epinephrine.

```
qqnorm(residuals(dogsanova),main = "QQ plot of estimated erros")
```



QQ plot of residuals eik = Yik - mui (k = 1 to 10 and i = 1 to 3) looks close to a normal distribution.

```
plot(fitted(dogsanova),residuals(dogsanova))
```



The plot of fitted(group means) against residuals show no pattern.

Therefore, assumptions for one-way ANOVA hold true.

Estimated concentrations of epinephrine for different drugs:

```
#isoflurane
mean(dogs[,1])
```

```
## [1] 0.434
```

```
#halothane
mean(dogs[,2])
```

```
## [1] 0.469
```

```
#cyclopropane
mean(dogs[,3])
```

```
## [1] 0.853
```

Estimated average concentrations of epinephrine for isoflurane, halothane and cyclopropane are 0.434,0.469 and 0.853 respectively.

3d.

Krushal-Wallis Test:

```
kruskal.test(yield~variety,data=dogsframe)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  yield by variety
## Kruskal-Wallis chi-squared = 5.6442, df = 2, p-value = 0.05948
```

p-value for krushal-wallis test comes out to be 0.05948. Therefore, HA: mu1 = mu2 = mu3 can't be rejected. The type of drug has no effect on the concentration of plasma epinephrine.

Conclusion - The reliability of ANOVA test depends on the assumption that the data comes from a normal dsitribution. Whereas, Krushal-Wallis Test is a non-paramteric test and doesn't depedent on the distribution of data. For the above data ANOVA test and Krushal-Wallis Test arrive at different conclusions. ANOVA test accepts that the type of drug has an effect on the concentration of plasma epinephrine whereas Krushal-Wallis test rejects this claim. The above data follows a normal distribution therefore, the result of one-way ANOVA is more reliable than Krushal-Wallis Test.

Exercise 4

4a.

To randomize the data, we run:

```
data=read.table(file="hemoglobin.txt",header=TRUE)
data_random = data[sample(nrow(data)),]
```

4b.

```
aov_model = aov(hemoglobin ~ rate * method, data=data_random)
summary(aov_model)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## rate         1  27.93  27.931  11.933 0.000905 ***
## method       1   2.42   2.415   1.032 0.312963
## rate:method  1   1.24   1.243   0.531 0.468373
## Residuals   76 177.90   2.341
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see in the summary that the variables *rate*, *method* and their interaction, *rate:method*, all have p-values lower than 0.05. This means that all these variables are statistically significant, and thus influence the hemoglobin levels.

4c.

We can see that the p-value for *rate* significantly lower than the p-values for *method* and *rate:method*,with a factor of 1000. This seems to indicate that *rate* has a lot more influence on the hemoglobin levels than the other variables. However, the interaction *rate:method} falls within statistical significance. This means that the interaction between the variables influences the hemoglobin levels, and whether one or the other has more influence is not really relevant.

The combination of rate 2 with method B yields the highest hemoglobin levels, which is 11.7. We filter the dataset on only rate 3 and method A, and then we compute the mean of the hemoglobin column of the filtered dataset:

```
rate_3 = subset(data_random, rate==3)
rate_3_A = subset(rate_3, method=="A")
mean = mean(rate_3_A$hemoglobin)
print(paste0("Mean of rate 3 method A: ", mean))
```

```
## [1] "Mean of rate 3 method A: 9.03"
```

We then compute the means for each rate:

```
rate_1 = subset(data_random, rate==1)
rate_2 = subset(data_random, rate==2)
rate_4 = subset(data_random, rate==4)

mean_1 = mean(rate_4$hemoglobin)
mean_2 = mean(rate_4$hemoglobin)
mean_3 = mean(rate_4$hemoglobin)
mean_4 = mean(rate_4$hemoglobin)

print(paste0("Mean of rate 1: ", mean_1))
```

```
## [1] "Mean of rate 1: 8.855"
```

```
print(paste0("Mean of rate 2: ", mean_2))
```

```
## [1] "Mean of rate 2: 8.855"
```

```
print(paste0("Mean of rate 3: ", mean_3))
```

```
## [1] "Mean of rate 3: 8.855"
```

```
print(paste0("Mean of rate 4: ", mean_4))
```

```
## [1] "Mean of rate 4: 8.855"
```

This means that **rate 2** gives the highest mean hemoglobin level.

4d.

We perform a one-way ANOVA test with:

```
anova_one = aov(hemoglobin ~ rate, data=data_random)
summary(anova_one)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## rate         1  27.93  27.931      12 0.000867 ***
## Residuals   78 181.55   2.328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that the p-value is below 0.05, which means that we can reject the null-hypothesis. This means that that the hemoglobin level differs for the various rates. For a one-way ANOVA test to work, there need to be at least 3 different categories. Thus, for the rate variable, this is a useful test. It would not be a useful test if we wanted to use the method variable.

Exercise 5

5a.

```
cream <- read.csv("cream.txt", sep="")
is.vector(cream$acidity)
```

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

```
cream$batch = as.factor(cream$batch)
cream$position = as.factor(cream$position)
cream$starter = as.factor(cream$starter)

creamanov = lm(acidity~batch+position+starter,data=cream)
anova(creamanov)
```

```
## Analysis of Variance Table
##
## Response: acidity
##           Df Sum Sq Mean Sq F value    Pr(>F)
## batch      4 18.778  4.6944  8.5975  0.001632 **
## position   4  2.348  0.5870  1.0750  0.411191
## starter    4 44.136 11.0340 20.2080 2.904e-05 ***
## Residuals 12  6.552  0.5460
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(creamanov)
```

```
##
## Call:
## lm(formula = acidity ~ batch + position + starter, data = cream)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2836 -0.2336  0.0384  0.3584  1.0204
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.6616     0.5329  16.255 1.55e-09 ***
## batch2       -1.3480     0.4673  -2.884   0.0137 *
## batch3        0.2760     0.4673   0.591   0.5658
## batch4        1.3680     0.4673   2.927   0.0127 *
## batch5        0.2000     0.4673   0.428   0.6763
```

```
## position2    -0.6180     0.4673  -1.322   0.2107
## position3    -0.0380     0.4673  -0.081   0.9365
## position4    -0.7640     0.4673  -1.635   0.1280
## position5    -0.2640     0.4673  -0.565   0.5825
## starter2     -0.1500     0.4673  -0.321   0.7538
## starter3     -0.9800     0.4673  -2.097   0.0579 .
## starter4      2.8100     0.4673   6.013 6.10e-05 ***
## starter5     -0.4840     0.4673  -1.036   0.3208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7389 on 12 degrees of freedom
## Multiple R-squared:  0.9088, Adjusted R-squared:  0.8175
## F-statistic:  9.96 on 12 and 12 DF,  p-value: 0.0001777
```

From the three way experiment above we can see that there is a difference between the effects of different starter values on the acidity in the column $Pr(>|t|)$ which shows that there is a difference between starter1 and 2 of 0.7538 which is much larger than 0.05 which is generally used as the cutoff for significant effects. This means that there is a large difference between starter1 which is very significant and starter2 which is not significant.

5b.

From the test results seen above in the subsection for 5a we can see that all the values for the position are above the 0.05 cutoff of significance meaning it has no real significance on the acidity and can be left out from the test which will look like the results below.

```
creamanov = lm(acidity~batch+starter,data=cream)
anova(creamanov)
```

```
## Analysis of Variance Table
##
## Response: acidity
##           Df Sum Sq Mean Sq F value    Pr(>F)
## batch      4 18.778  4.6944  8.4392 0.0007348 ***
## starter    4 44.136 11.0340 19.8360 4.816e-06 ***
## Residuals 16  8.900  0.5563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(creamanov)
```

```
##
## Call:
## lm(formula = acidity ~ batch + starter, data = cream)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5648 -0.2548 -0.0548  0.3592  1.1352
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.3248     0.4475  18.603 2.91e-12 ***
## batch2       -1.3480     0.4717  -2.858   0.0114 *
## batch3        0.2760     0.4717   0.585   0.5666
## batch4        1.3680     0.4717   2.900   0.0104 *
```

```
## batch5        0.2000     0.4717   0.424   0.6772
## starter2     -0.1500     0.4717  -0.318   0.7546
## starter3     -0.9800     0.4717  -2.078   0.0542 .
## starter4      2.8100     0.4717   5.957 2.01e-05 ***
## starter5     -0.4840     0.4717  -1.026   0.3201
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7458 on 16 degrees of freedom
## Multiple R-squared:  0.8761, Adjusted R-squared:  0.8141
## F-statistic: 14.14 on 8 and 16 DF,  p-value: 6.474e-06
```

5c.

The friedman test can be used to test if none of the the factors have an effect on the acidity and so we only need to run it if we are not sure any of the factors have ann effect and is good to do just in case. The result of the Friedman test can be seen here.

```
friedman.test(cream$acidity, cream$batch, cream$starter,data=cream)
```

```
##
##  Friedman rank sum test
##
## data:  cream$acidity, cream$batch and cream$starter
## Friedman chi-squared = 13.12, df = 4, p-value = 0.0107
```

From this we can see that with a p-value 0.0107 the H0 hypothesis that there is no effect on the acidity is rejected.

5d.

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
creamlmer = lmer(acidity~(1|batch)+starter,data=cream, REML=FALSE)
summary(creamlmer)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: acidity ~ (1 | batch) + starter
##    Data: cream
##
##      AIC      BIC   logLik deviance df.resid
##     75.4     83.9    -30.7     61.4       18
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3633 -0.5283 -0.1047  0.5699  1.7196
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  batch    (Intercept) 0.6621   0.8137
##  Residual             0.4450   0.6671
## Number of obs: 25, groups:  batch, 5
##
## Fixed effects:
```

```
##             Estimate Std. Error t value
## (Intercept)   8.4240     0.4706  17.902
## starter2     -0.1500     0.4219  -0.356
## starter3     -0.9800     0.4219  -2.323
## starter4      2.8100     0.4219   6.660
## starter5     -0.4840     0.4219  -1.147
##
## Correlation of Fixed Effects:
##          (Intr) strtr2 strtr3 strtr4
## starter2 -0.448
## starter3 -0.448  0.500
## starter4 -0.448  0.500  0.500
## starter5 -0.448  0.500  0.500  0.500
```

```
creamlmer1 = lmer(acidity~(1|batch),data=cream, REML=FALSE)
anova(creamlmer, creamlmer1)
```

```
## Data: cream
## Models:
## creamlmer1: acidity ~ (1 | batch)
## creamlmer: acidity ~ (1 | batch) + starter
##            npar    AIC     BIC  logLik deviance  Chisq Df Pr(>Chisq)
## creamlmer1    3 103.07 106.724 -48.534   97.068
## creamlmer     7  75.37  83.902 -30.685   61.370 35.698  4  3.339e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the summary of the mixed effect analysis we can see in the $Pr(> Chisq)$ column of the anova test, which can in effect be viewed as the p-value of the starter factor sits at a very low value of 3.339e-07 making it very significant. When compared to the p-value of the anova test in section 5b which had a p-value of 4.816e-06 for the starter factor the difference is there but it is very minor with the mixed effect analysis achieving a slightly lower p-value