# Assignment 1

Andrea Di Dio, Chenghan Song, Jiacheng Lu — Group 22

Exercise 1

a)

We generate two samples of sizes 30 from a standard normal distribution and programme loops to calculate the power of the t-test for every value of nu:

```
n=m=30;mu=180;sd=5;B=1000;p=numeric(B);
nu=seq(175,185,by=0.25);
power=vector();
for(i in 1:41){ 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[i]=mean(p<0.05)
}
plot(nu,power,type="p");
```
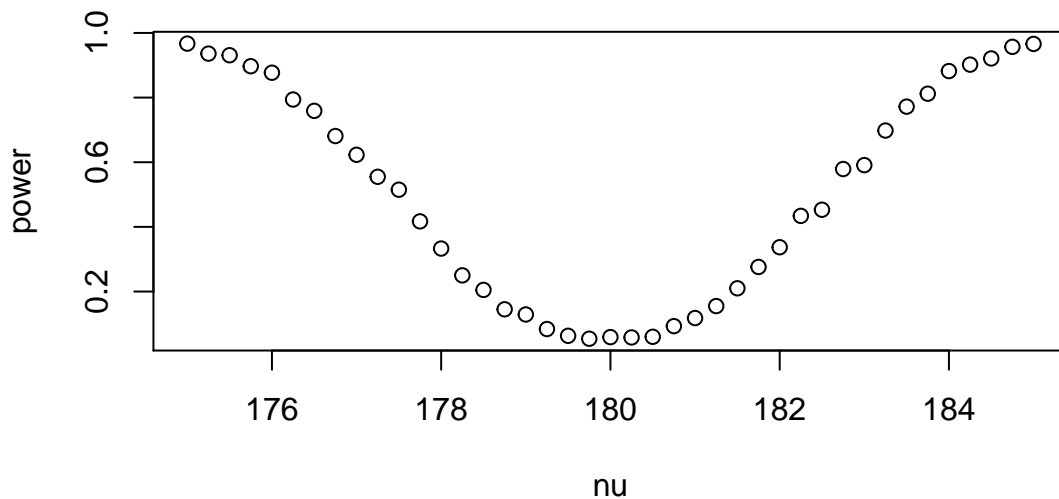

b)

Change n and m to 30:

```
p2=numeric(B); power2=vector(); n2=m2=100;
for(i in 1:41){ for(b in 1:B){ x2=rnorm(n2,mu,sd); y2=rnorm(m2,nu[i],sd);
    p2[b]=t.test(x2,y2,var.equal = TRUE)[[3]]}
  power2[i]=mean(p2<0.05);
}
plot(nu,power,type="p"); lines(nu,power2,type="p",col="red");
```
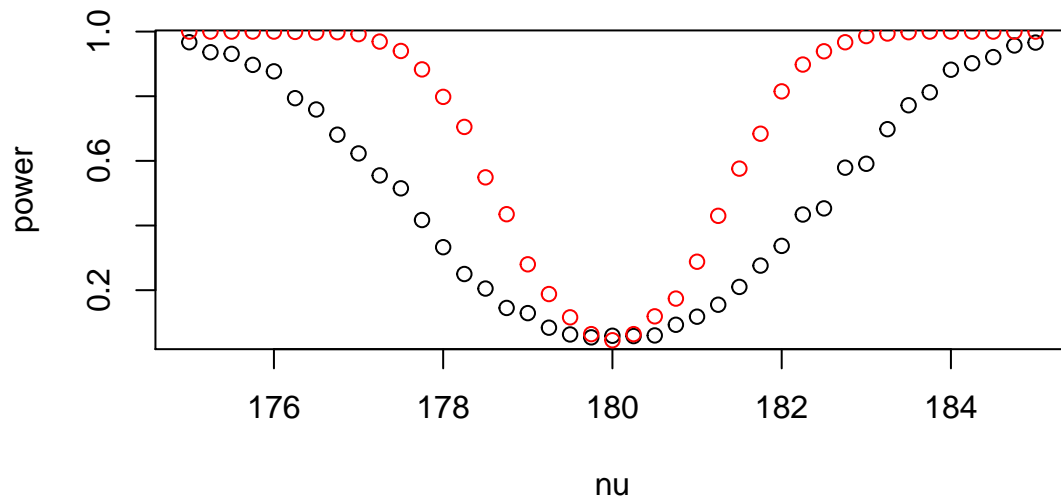
c)

Change sd to 15:

```
sd3=15;p3=numeric(B);
power3=vector();
for(i in 1:41){ for(b in 1:B){x3=rnorm(n,mu,sd3); y3=rnorm(m,nu[i],sd3);
    p3[b]=t.test(x3,y3,var.equal = TRUE)[[3]]}
  power3[i]=mean(p3<0.05);
}
plot(nu,power,type="p"); lines(nu,power2,type="p",col="red"); lines(nu,power3,type="p",col="blue");
```
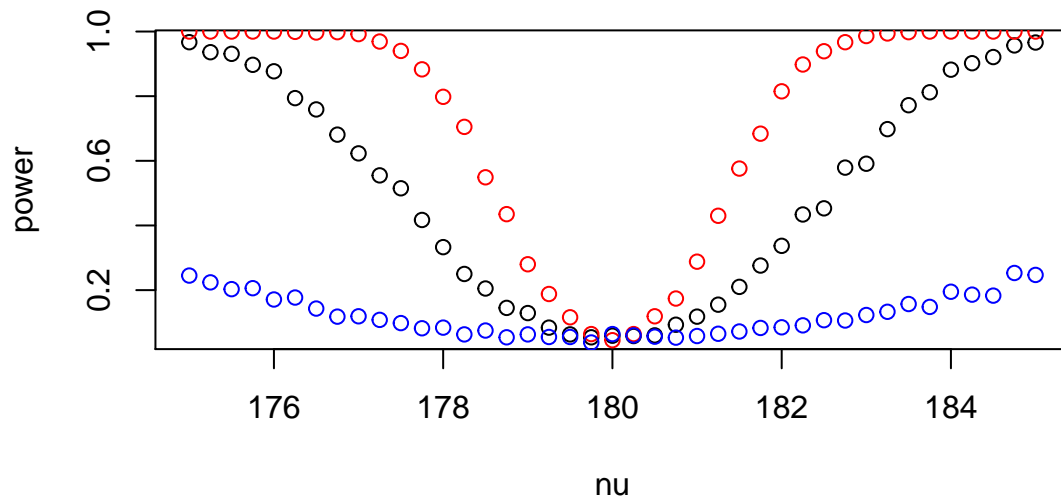

d)

As in a), we can see that if nu gets closer to 180 which equals mu, the power of t-test gets smaller. Compare b) with a), we can find out that if we enlarge the size of samples, the nu has to be more closer to 180 to get a low power. It is more accurate. When it comes to c), we can see that with the growth of sd, the rate of the change of power curve becomes smaller, which means the probability that t-test rejects the null hypothesis becomes smaller.

Exercise 2

a)

$H_0 : \mu = \mu_0$ (population is normally distributed)

```r
light79_raw <- read.delim("./light1879.txt", header = FALSE, sep = "")
light82_raw <- read.delim("./light1882.txt", header = FALSE, fill = TRUE, sep = "")
light_raw <- read.delim("./light.txt", header = FALSE, fill = TRUE, sep = "")
light79 <- unlist(na.omit(light79_raw) + 299000, use.name = FALSE)
light82 <- unlist(na.omit(light82_raw) + 299000, use.name = FALSE)
light <- unlist((7.442 / ((na.omit(light_raw)/1000) + 24.8) * 1000000), use.names = FALSE)

par(mfrow = c(1,3))
hist(light79); boxplot(light79, main = "Box plot of light79")
qqnorm(light79, main = "Normal Q-Q plot of light79"); qqline(light79)
```
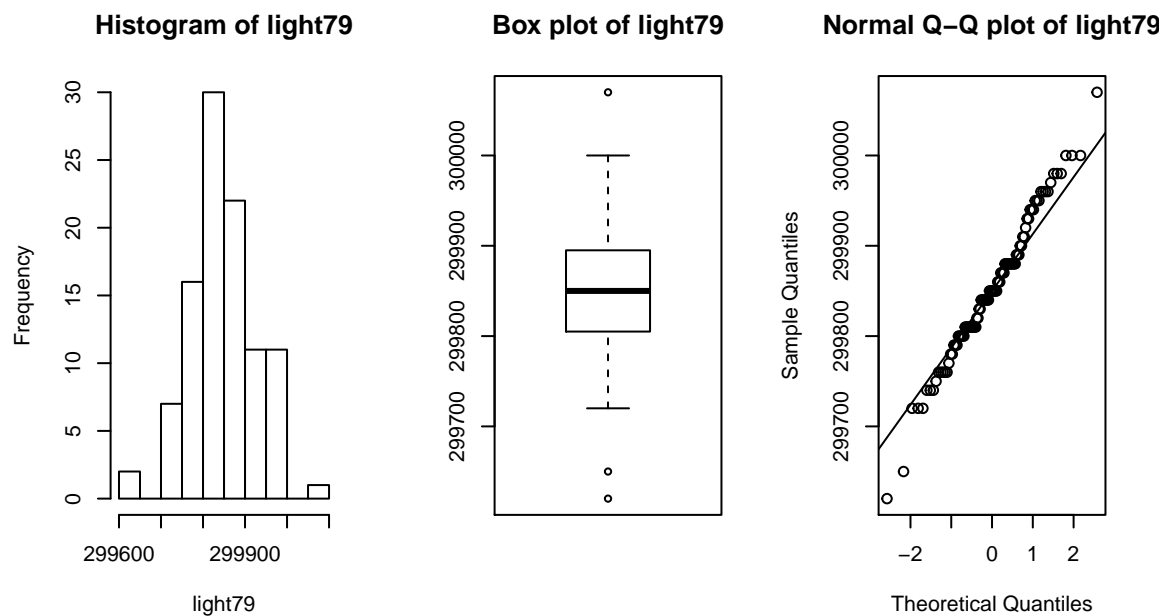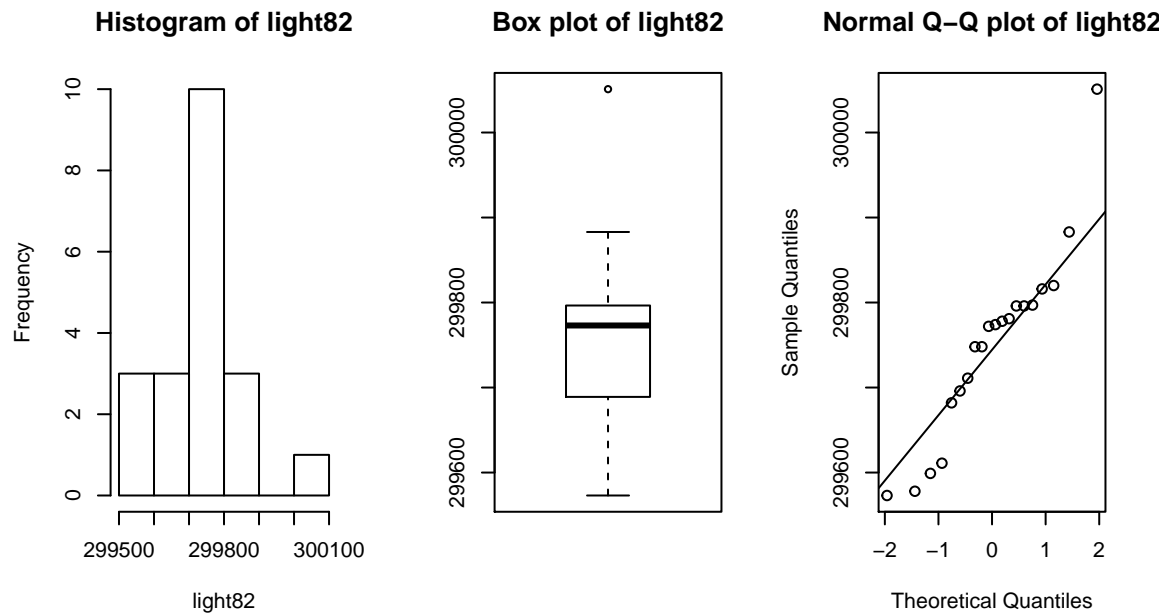


```r
shapiro.test(light79)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  light79
## W = 1, p-value = 0.5
```

```r
par(mfrow = c(1,3))
hist(light82); boxplot(light82, main = "Box plot of light82")
qqnorm(light82, main = "Normal Q-Q plot of light82"); qqline(light82)
```

3

Histogram of light82 **Box plot of light82** **Normal Q–Q plot of light82**


```
shapiro.test(light82)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  light82
## W = 0.9, p-value = 0.1
```

```
par(mfrow = c(1,3))
hist(light); boxplot(light, main = "Box plot of light")
qqnorm(light, main = "Normal Q-Q plot of light"); qqline(light)
```

Histogram of light **Box plot of light** **Normal Q–Q plot of light**
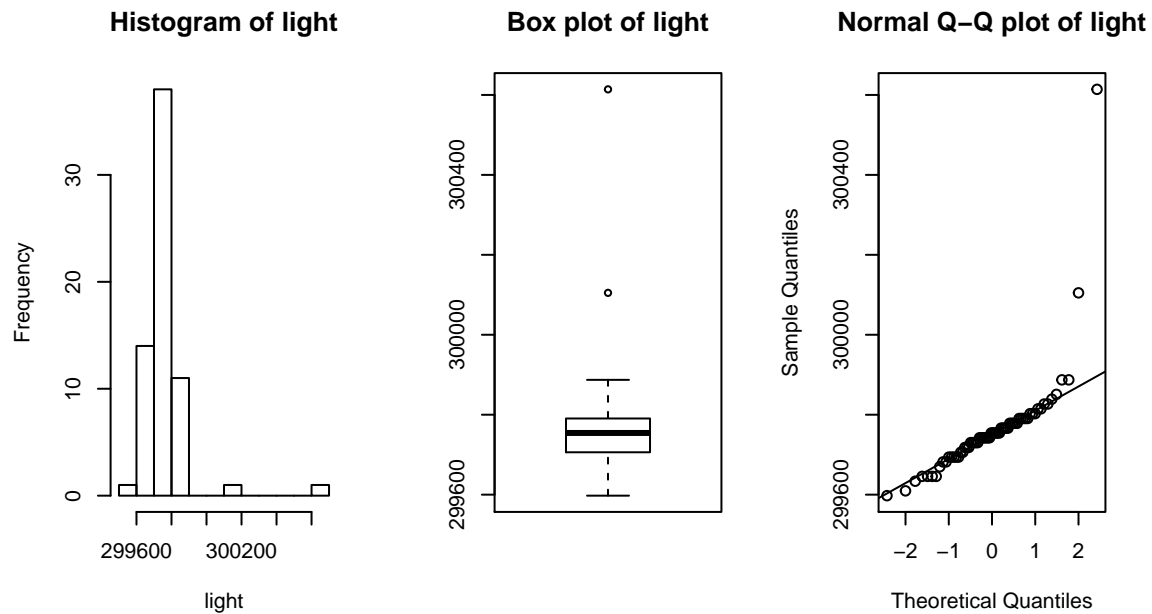

```
shapiro.test(light)
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  light
## W = 0.6, p-value = 3e-12
```

Visually, the histogram, boxplots and qq plots of the data sets might suggest that for light79 and light82 the population might be normally distributed. On the other hand, the histogram of light has some significant outliers which cause the QQ plot to show non-normal distribution. In order to assess the normality of these datasets, we have conducted the **Shapiro-Wilk normality test** which shows $p > \alpha$ assuming $\alpha = 0.05$ for the light79 and light82 datasets but $p < \alpha$ for the light dataset. This means that we *fail to reject H_0 for light79 and light82, but we can safely reject H_0 for light.*

b)

```
T1 <- mean(light79); T2 <- mean(light82); T3 <- mean(light)

conf_interval <- function(light_data, T_val) {
  B <- 1000; Tstar <- numeric(B)
  for(i in 1:B) { Xstar <- sample(light_data, replace = TRUE);
    Tstar[i] <- mean(Xstar) }
  Tstar25 <- quantile(Tstar, 0.025); Tstar975 <- quantile(Tstar, 0.975)
  return(c(2*T_val - Tstar975, 2*T_val - Tstar25))
}

light79_int <- conf_interval(light79, T1)
light82_int <- conf_interval(light82, T2)
light_int <- conf_interval(light, T3)
cat("Light 79: Interval = ", light79_int, "Mean = ", T1, "\n")
```

```
## Light 79: Interval =  3e+05 3e+05 Mean =  3e+05
```

```
cat("Light 82: Interval = ", light82_int, "Mean = ", T2, "\n")
```

```
## Light 82: Interval =  3e+05 3e+05 Mean =  3e+05
```

```
cat("Light: Interval = ", light_int, "Mean = ", T3, "\n")
```

```
## Light: Interval =  3e+05 3e+05 Mean =  3e+05
```

All the means calculated fall into the range of the 95% bootstrap confidence intervals calculated. The interval seems to be rather large, however, by using the population mean as the estimate statistic, we have a statistic that is highly sensitive to the outliers.

c)

```
current_c <- 299792 # https://www.space.com/15830-light-speed.html
# assuming normality for 79 and 82 we can use t-test, for light we need wilcoxon test
t.test(light79, mu = current_c)
```

```
##
##  One Sample t-test
##
## data:  light79
## t = 8, df = 99, p-value = 1e-11
## alternative hypothesis: true mean is not equal to 3e+05
## 95 percent confidence interval:
##  3e+05 3e+05
## sample estimates:
```

```
## mean of x
##     3e+05
```

```
t.test(light82, mu = current_c)
```

```
##
##   One Sample t-test
##
## data:  light82
## t = -2, df = 19, p-value = 0.1
## alternative hypothesis: true mean is not equal to 3e+05
## 95 percent confidence interval:
##   3e+05 3e+05
## sample estimates:
## mean of x
##     3e+05
```

```
wilcox.test(light, mu = current_c)
```

```
##
##   Wilcoxon signed rank test with continuity correction
##
## data:  light
## V = 387, p-value = 4e-06
## alternative hypothesis: true location is not equal to 3e+05
```

H_0: The true mean of the speed of light for the datasets is c

By assuming normality for the datasets light79 and light82, we can run a **one sample t-test** to test if the current speed of light is consistent with our measurements. For the light dataset, we will use a **Wilcoxon signed rank test**. For light79, $p < \alpha$ meaning that we can reject H_0. For light82, $p > \alpha$ and therefore we fail to reject H_0. For light, $p < \alpha$, thus we can safely reject H_0.

Exercise 3

a)

We can make a histiogram and a QQ-plot to show the data set. From the histogram and QQ-plot of the first month bills, we can see that most subscribers' bills is between 0 to 20; bills more than 70 are also of a large quantity. So my advice for the manager is to set two kinds of month bills: one is between 0 to 20, the other is more than 70. We don't think there is any inconsistency in the data. For 0, people may just don't use mobile phones.

```
data=read.table(file="telephone.txt",header=TRUE)
par(mfrow = c(1,2))
hist(data$Bills,prob=T); qqnorm(data$Bills, main = "QQ plot of data$Bills")
```

Histogram of data$Bills
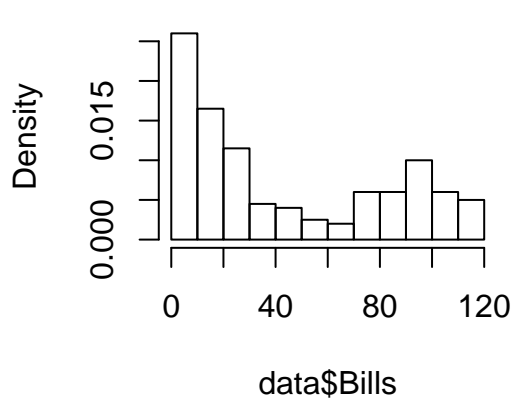
QQ plot of data$Bills
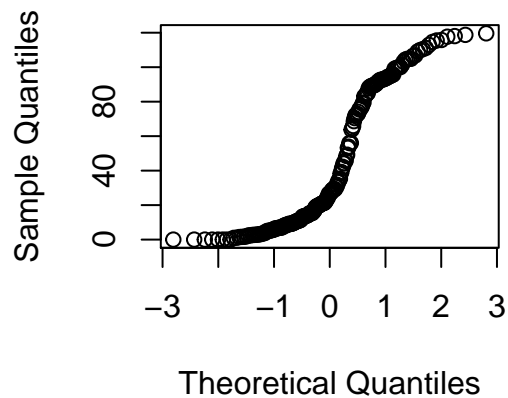
b)

In order to test lamda from[0.01,0.1],let lamda=seq(0.01,0.1,by=0.01);for every lamda, use bootstrap test.

```r
lamda=seq(0.01,0.1,by=0.01); length(lamda);bills=data$Bills
```

```
## [1] 10
```

```r
t=median(bills);
B=1000; tstar=numeric(B)
p=numeric(10); n=length(bills)
for(i in 1:10){ for (j in 1:B){
    xstar=rexp(n,lamda[i]); tstar[j]=median(xstar)}
  pl=sum(tstar<t)/B; pr=sum(tstar>t)/B
  p[i]=2*min(pl,pr)}
p
```

```
##  [1] 0.000 0.010 0.146 0.000 0.000 0.000 0.000 0.000 0.000 0.000
```

We can easily find out that p[3] is bigger than 0.05, which means when data$Bills stems from the exponential distribution Exp(0.03);

c)

Confidence 1-2*alpha=0.95, according to the formula its confidence interval should be $[2\,T\text{-}Tstar(1\text{-}alpha),2\,T\text{=}Tstar(alpha)]$;

```r
bills=data$Bills; B=1000
Tstar=numeric(B); T1=median(bills)
for(i in 1:B){Xstar=sample(bills,replace = TRUE);
  Tstar[i]=median(Xstar)}
Tstar25=quantile(Tstar,0.025); Tstar975=quantile(Tstar,0.975)
sum(Tstar<Tstar25); c(2*T1-Tstar975,2*T1-Tstar25)
```

```
## [1] 25
```

```
## 97.5%  2.5%
##  19.5  33.6
```

d)

Central limit theorem establishes that, in some situations, when independent random variables are added, their properly normalized sum tends toward a normal distribution even if the original variables themselves are not normally distributed. We can set up 1,000 sets of samples, each group sampling 50 data. The mean of 1000 samples is close to the population mean. According to the exponential distribution, mean=1/lamda, so lamda is about 0.022.

```
B=1000; s=numeric(B)
for(i in 1:B){ sample=sample(bills,50); sample_mean=mean(sample)
  s[i]=sample_mean}
s_mean=mean(s); s_var=var(s)
lamda=1/s_mean;
lamda
```

[1] 0.0229

According to the above distribution,rexp(n,lamda):

```
B=1000;
Tstar=numeric(B); T1=median(bills)
for(i in 1:B){Xstar=sample(rexp(n,lamda));
  Tstar[i]=median(Xstar)}
Tstar25=quantile(Tstar,0.025); Tstar975=quantile(Tstar,0.975)
sum(Tstar<Tstar25); c(2*T1-Tstar975,2*T1-Tstar25)
```

[1] 25

97.5% 2.5%
17.2 29.4

e)

to check whether the fraction of the bills less than 10 euro is at most 25%. Null hypothesis: median bill >= 40 euro. Choose sign test. The result shows p-value = 0.009698, which is smaller than 0.05, so reject H0, the median bill is smaller than 40 euro.

```
binom.test(sum(bills>40),200,al="l");
```

```
##
##  Exact binomial test
##
## data:  sum(bills > 40) and 200
## number of successes = 83, number of trials = 200, p-value = 0.01
## alternative hypothesis: true probability of success is less than 0.5
## 95 percent confidence interval:
##  0.000 0.475
## sample estimates:
## probability of success
##                  0.415
```

Null hypothesis: the fraction of bills less than 10 euro is at most 25%; p-value=0.3983,the fraction of the bills less than 10 euro is at most 25%.

```
low=sum(bills<10);
binom.test(low,200,p=0.25,alternative="greater",conf.level = 0.9)
```

```
##
##  Exact binomial test
##
## data:  low and 200
```

```
## number of successes = 52, number of trials = 200, p-value = 0.4
## alternative hypothesis: true probability of success is greater than 0.25
## 90 percent confidence interval:
##  0.22 1.00
## sample estimates:
## probability of success
##                   0.26
```
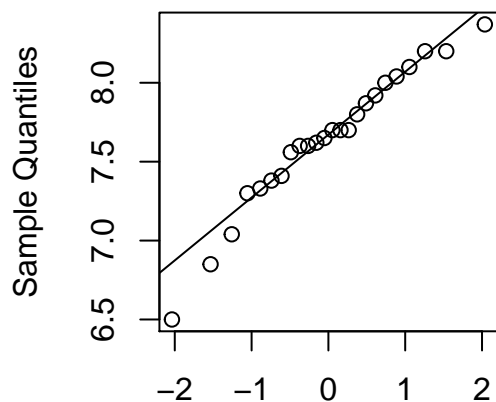
Exercise 4

a)

```r
run_raw <- read.table("run.txt", header = TRUE)
before <- run_raw$before; after = run_raw$after
cor.test(before, after)
```

```
##
##  Pearson's product-moment correlation
##
## data:  before and after
## t = 4, df = 22, p-value = 8e-04
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.317 0.829
## sample estimates:
##    cor
## 0.639
```

```r
par(mfrow = c(1,2))# Check normality assumption
qqnorm(before, main = "QQ plot of runtime before"); qqline(before)
qqnorm(after, main = "QQ plot runtime after"); qqline(after)
```
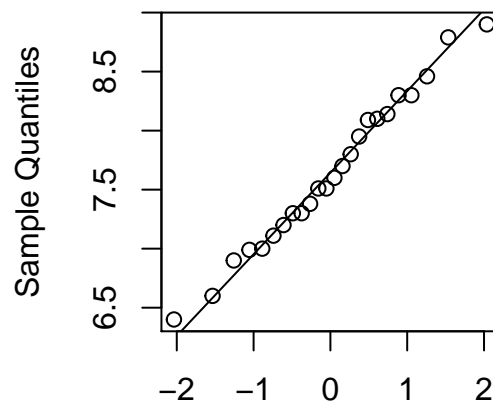

Pearson's Correlation test shows that there is significant correlation between the runtimes before and after the drink. This test assumes normality, which looking at the QQ plots for the runtimes of before and after can reasonably be assumed.

b)

```r
lemo <- split(run_raw, f = run_raw$drink)["lemo"]
energy <- split(run_raw, f = run_raw$drink)["energy"]

permutation_test <- function(x, y) {
  B <- 1000; Tstar <- numeric(B)
  for(i in 1:B) { Xstar <- t(apply(cbind(x, y), 1, sample))
    Tstar[i] <- mean(Xstar[,1] - Xstar[,2])}
  t <- mean(x - y); p1 <- sum(Tstar < t) / B; pr <- sum(Tstar > t) / B
  p <- 2 * min(p1, pr); return(p)
}

par(mfrow = c(1,2)); lemo_diff <- lemo$lemo$before - lemo$lemo$after;
energy_diff <- energy$energy$before - energy$energy$after
qqnorm(lemo_diff, main = "QQ plot of lemo difference"); qqline(lemo_diff)
qqnorm(energy_diff, main = "QQ plot of energy difference"); qqline(energy_diff)
```


QQ plot of lemo difference — **QQ plot of energy difference**
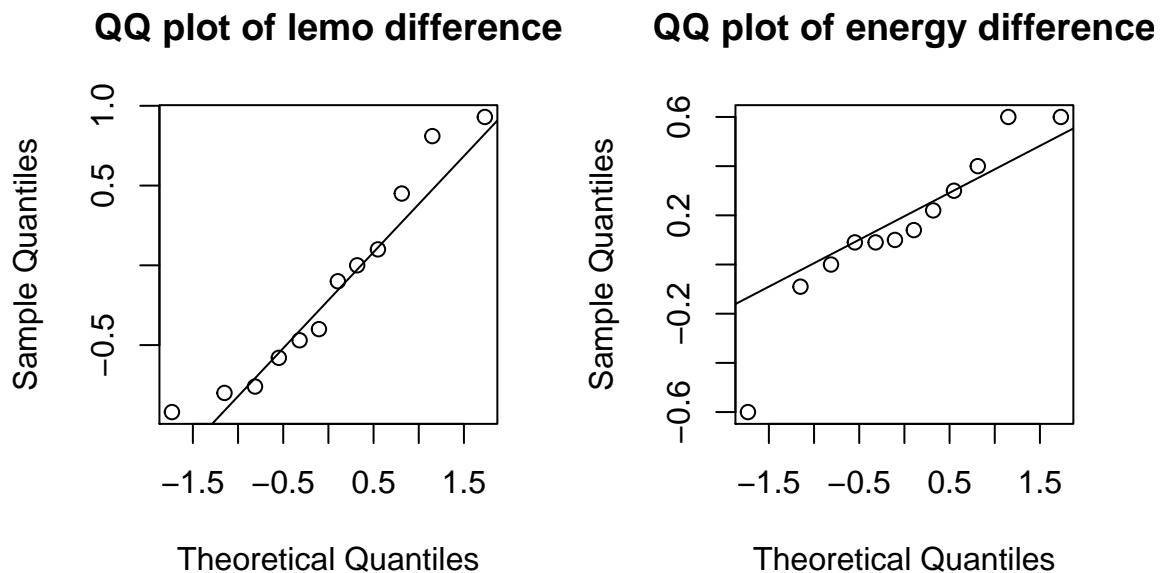
```r
p_lemo <- permutation_test(lemo$lemo$before, lemo$lemo$after)
p_energy <- permutation_test(energy$energy$before, energy$energy$after)
cat("p_lemo = ", p_lemo, " --- p_energy = ", p_energy, "\n")
```

```
## p_lemo =  0.428  --- p_energy =  0.13
```

H_0: There is no difference in speeds in the two running tasks.

Looking at the QQ plots for the differences in runtimes for the two types of drinks, we cannot assume normality and hence use a permutation test. for both the energy drink and the lemo drink, $p > \alpha$. Therefore, we fail to reject H_0.

c)

H_0: There is no difference between the two populations.

Looking at the QQ plots in b), we can assume that the differences are not normally distributed, hence, we perform a **Mann-Whitney test**.

```
wilcox.test(lemo_diff, energy_diff)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  lemo_diff and energy_diff
## W = 46, p-value = 0.1
## alternative hypothesis: true location shift is not equal to 0
```

The test results in $p > \alpha$ meaning that we fail to reject H_0.

d)

For both drinks, the sample size is too small to abide by the Central Limit Theorem, which requires $N \geq 30$, therefore meaning that we cannot assume that the population is normally distributed. In addition, there is no data regarding the amount of drink between the two runs, which for the experiment to remain fair should be the same quantity for each pupil, regardless of the type of drink. The tests in this exercise show that the two runtimes are correlated and we can assume that the two samples come from a similar population.
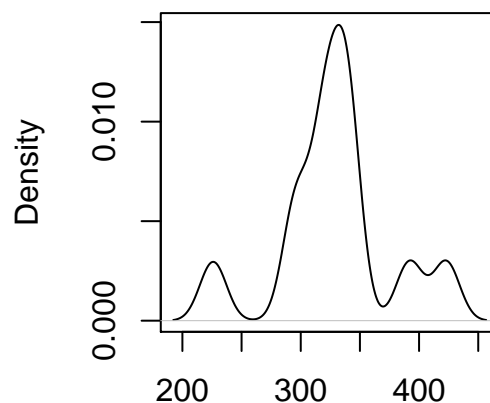
Exercise 5

a)

```
data("chickwts")
meatmealChick <- chickwts$weight[chickwts$feed == 'meatmeal']
sunflowerChick <- chickwts$weight[chickwts$feed == 'sunflower']
par(mfrow = c(1,2))
plot(density(meatmealChick))
plot(density(sunflowerChick))
```

density.default(x = meatmealCh density.default(x = sunflowerCh
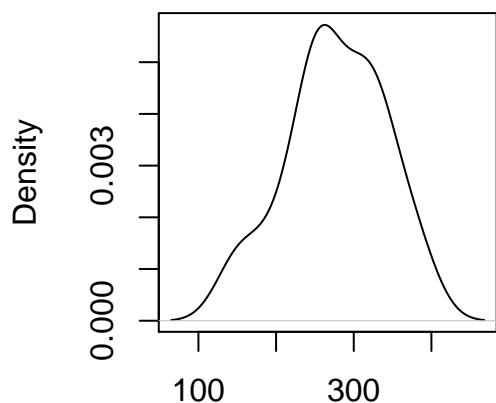
N = 11   Bandwidth = 29.31   N = 12   Bandwidth = 11.24

```
data("chickwts")
par(mfrow = c(1,1))
t.test(meatmealChick, sunflowerChick, var.equal = T)
```

```
##
##  Two Sample t-test
```

```
## 
## data:  meatmealChick and sunflowerChick
## t = -2, df = 21, p-value = 0.04
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -101.53   -2.48
## sample estimates:
## mean of x mean of y
##       277       329
```

p-value = 0.04047 < 0.05, so we reject H0 and conclude that a significant difference does exist, which means the mean chicken weights of meatmeal and sunflower groups are different.

It requires both types of weights should be normally distributed and Variances between two groups be equal. According to density plots, the distribution of sunflower is non-normal distribution, so the two-sample t-test is invalid.

```
wilcox.test(meatmealChick, sunflowerChick)
```

```
## 
##  Wilcoxon rank sum test
## 
## data:  meatmealChick and sunflowerChick
## W = 36, p-value = 0.07
## alternative hypothesis: true location shift is not equal to 0
```

The p-value is 0.06882 > 0.05, If the p-value is larger than 0.05, we cannot conclude that a significant difference exists. The conclusion is that the chicken weights of meatmeal and sunflower groups share the same median.

We require the two data samples are independent and the samples do not affect each other. So in this case, the test is valid.

```
ks.test(meatmealChick, sunflowerChick)
```

```
## 
##  Two-sample Kolmogorov-Smirnov test
## 
## data:  meatmealChick and sunflowerChick
## D = 0.5, p-value = 0.1
## alternative hypothesis: two-sided
```

The p-value is 0.1085 > 0.05, the chicken weights of meatmeal and sunflower groups should be identical and balanced in median, variability, and the shape of the distribution.

b)

```
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    15.4 5.9e-10 ***
## Residuals 65 195556    3009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value of the test is highly significant (p=5.94e-10), therefore we conclude the alternative, that at least one feed type has a different average weight. Hence the type of feed supplement has an effect on the weight of the chicks.

```
library("lattice")
xyplot(weight ~ feed, data=chickwts, main="", type=c("p","a"))
```
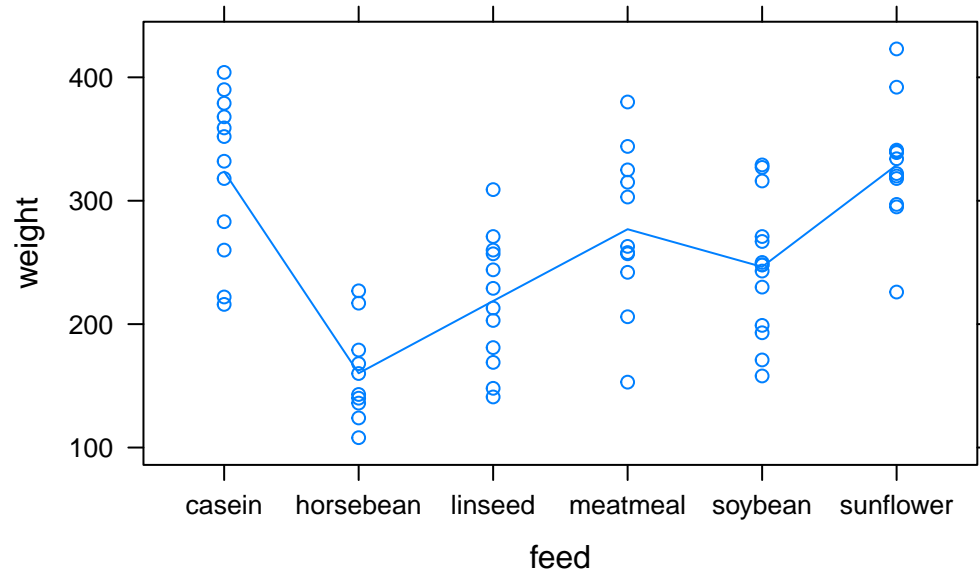

The plot above explains which feed types are producing the highest average weights.

```
summary(chickaov)
```

```
##
## Call:
## lm(formula = weight ~ feed, data = chickwts)
##
## 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 six supplements are: casein 323.6, horsebean: 160.2, linseed: 218.8, meatmeal: 276.9, soybean: 246.4, sunflower: 328.9.
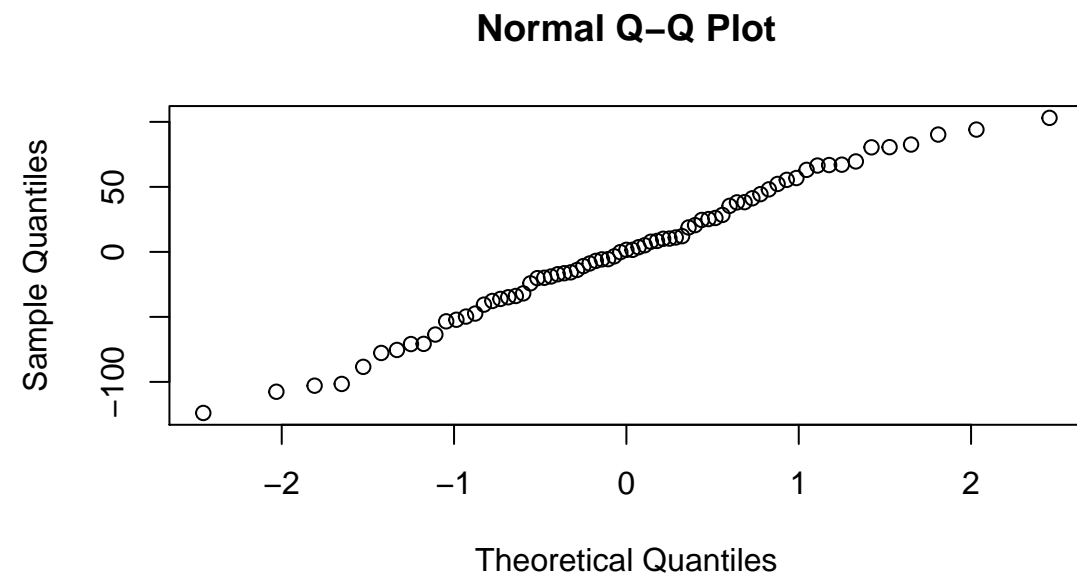
Sunflower and casein are the best feed supplements.

c)

```
opar <- par(mfrow = c(2, 2), oma = c(0, 0, 1.1, 0),
            mar = c(4.1, 4.1, 2.1, 1.1))
plot(chickaov)
```


lm(weight ~ feed)
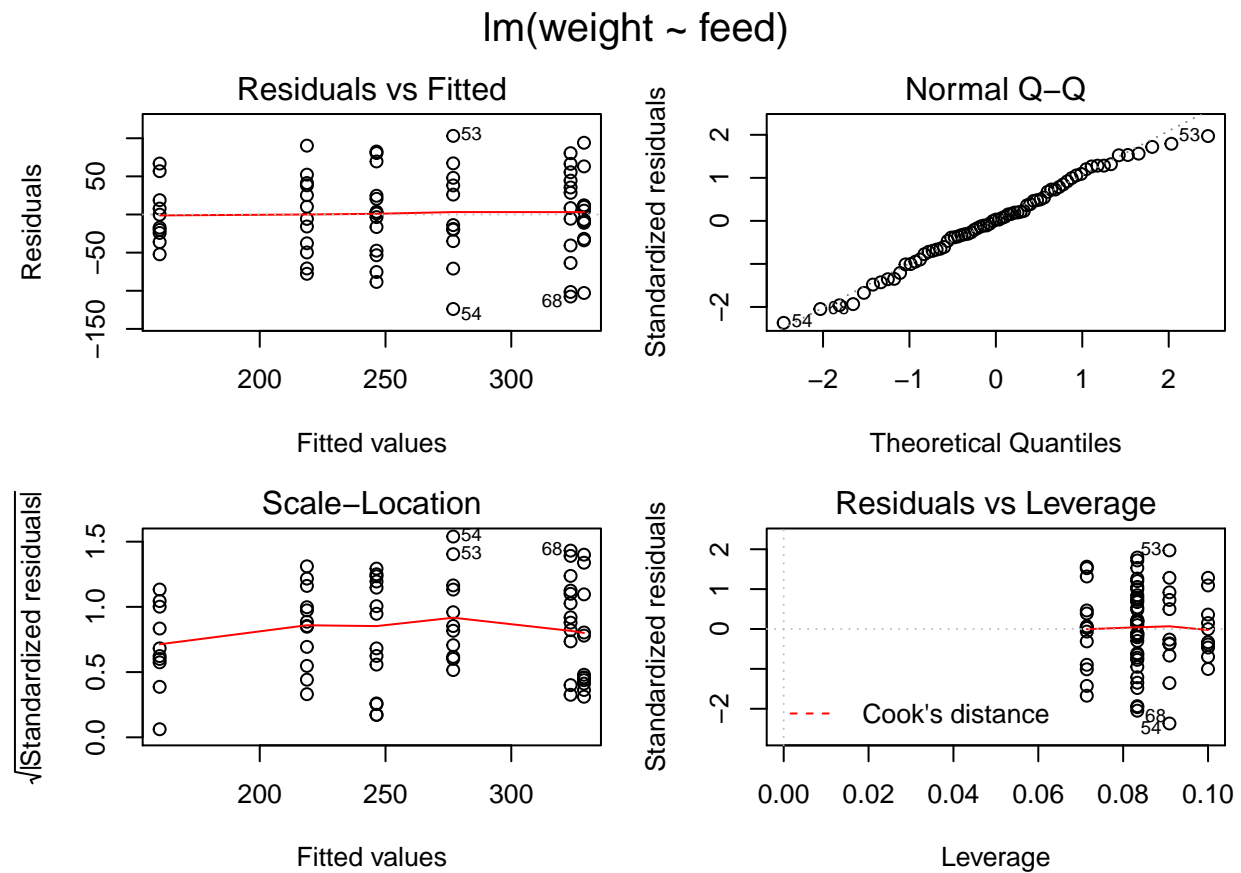
```
par(mfrow=c(1,1))
qqnorm(residuals(chickaov))
```


Normal Q–Q Plot

The plots demonstrate that the assumptions of ANOVA were satisfied as the residuals versus fitted values plot shows roughly constant variance, and the QQ-Plot shows normality of the residuals.

d)

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

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

The command kruskal.test performs the Kruskal-Wallis test and yields a p-value. The p-value for testing H0 is 5.113e-07, hence H0 is rejected. the type of feed supplement has an effect on the weight of the chicks.

The one-way ANOVA also yield a significant difference. From the qq plot above,the residuals do not seem to deviate significantly from normal, and both tests could be used here.