Assignment 3

Tanjina Islam, Miguel Morales Expósito and Carlos Perales Liñan, group 12

13 March 2018

Exercise 1

We load the data from the data source.

```
bread = read.table("bread.txt", header=TRUE)
attach(bread)
```

Task 1

For the randomization task, we define the variables N (the number of units for each combination), I (the levels of humidity) and J (the levels of environment). Then we perform the randomization.

```
N=3
I=2
J=3

rbind(rep(1:I, each = N*J), rep(1:J, N*I), sample(1:(N*I*J)))
```

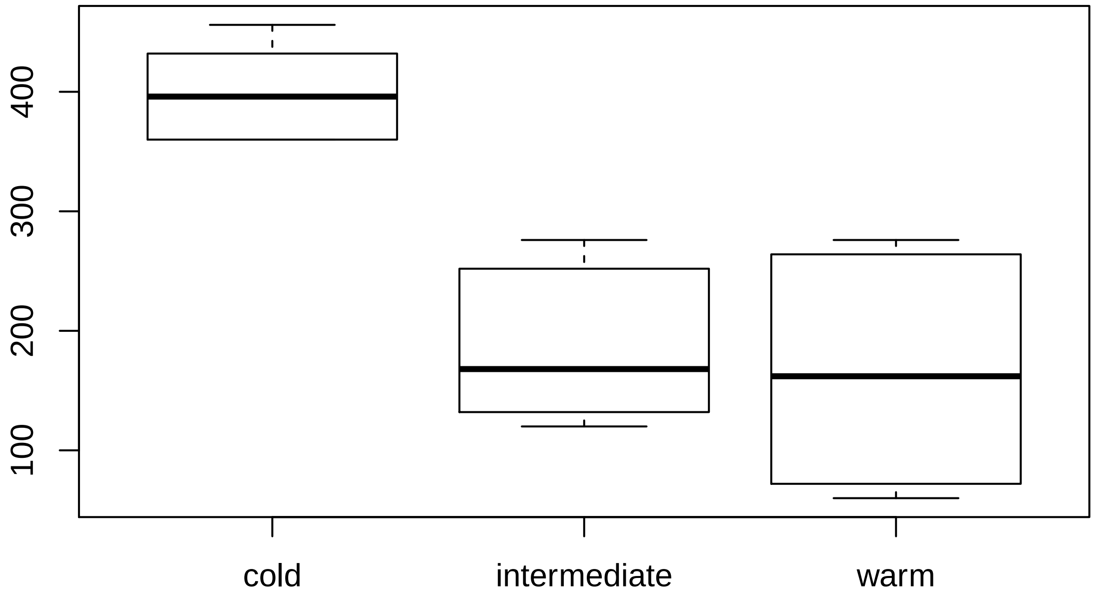
```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    1    1    1    1    1    1    1    1    1     2     2     2     2
## [2,]    1    2    3    1    2    3    1    2    3     1     2     3     1
## [3,]    1    3   11   18    5    9   15   16   13    14     4     8    17
##      [,14] [,15] [,16] [,17] [,18]
## [1,]     2     2     2     2     2
## [2,]     2     3     1     2     3
## [3,]    12     6     7     2    10
```

This way we can randomoly assign slices of bread to the different combination of the two factors.

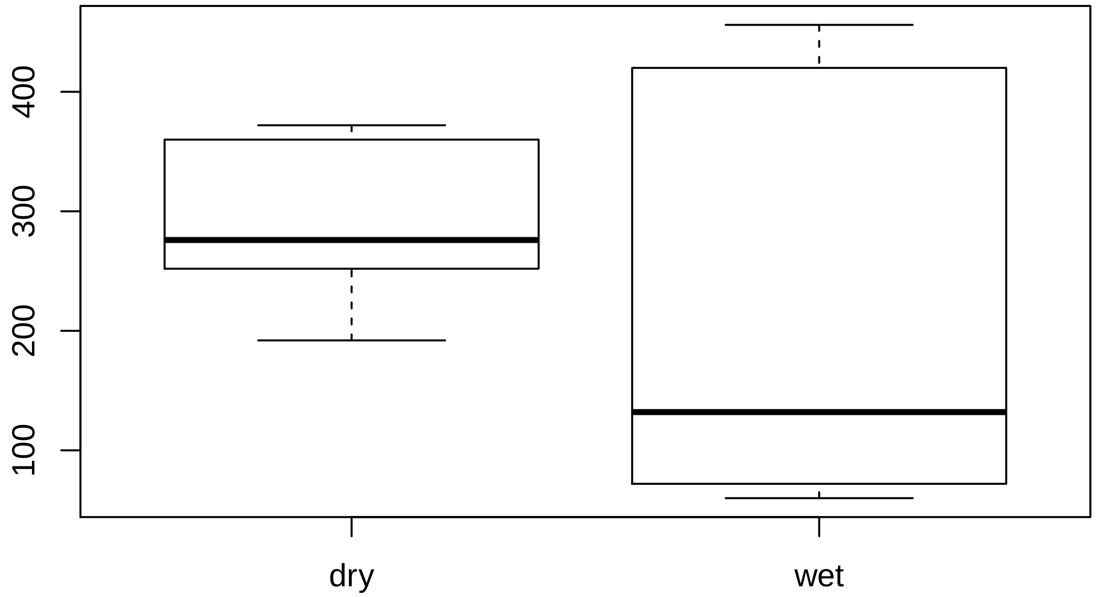
Task 2

First we make the 2 boxplots of hours versus the factors.

```
boxplot(hours~environment,data=bread)
```

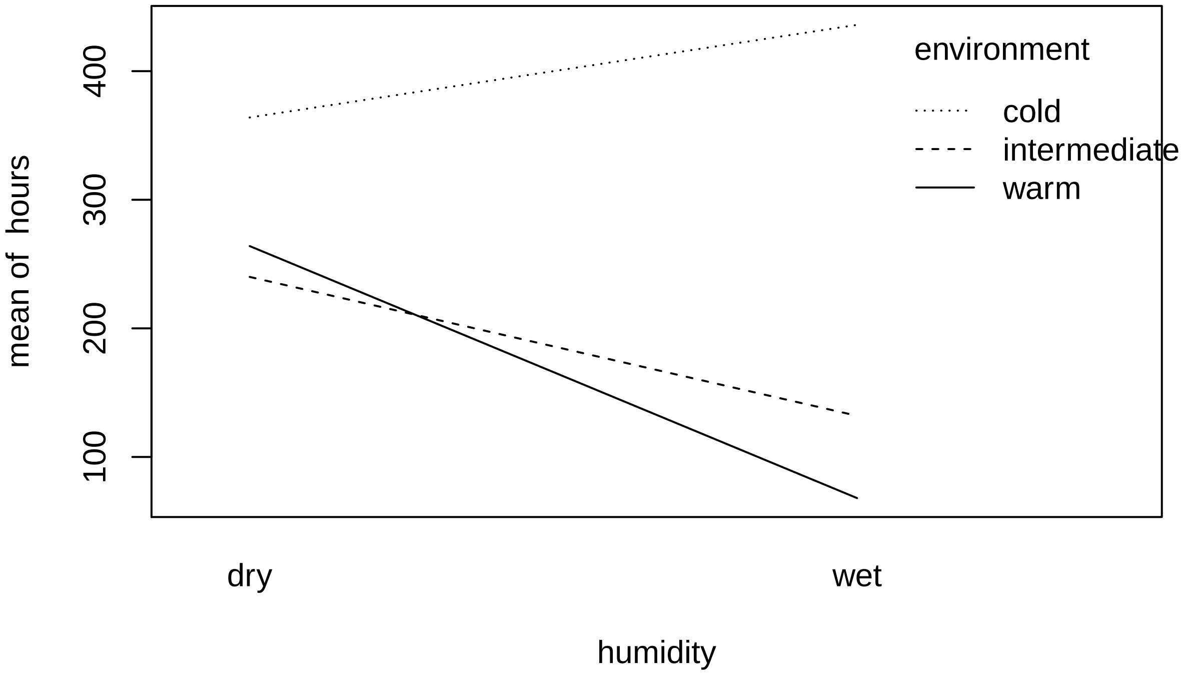


```
boxplot(hours~humidity,data=bread)
```

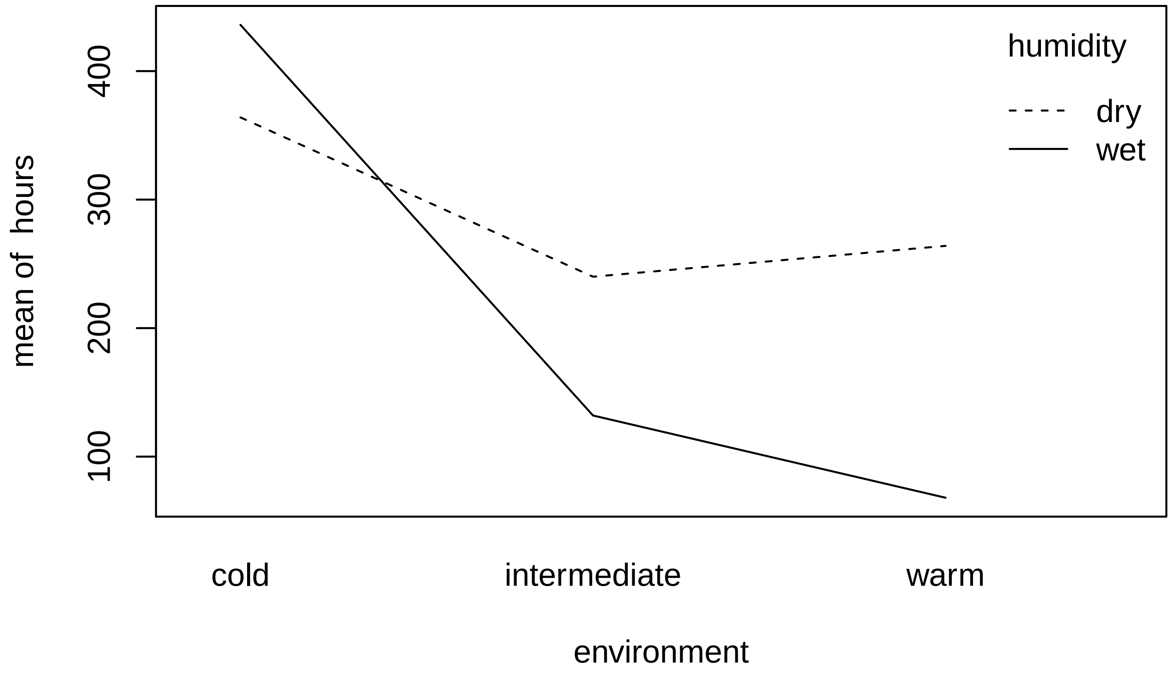


Next, we show the interaction plots against these same factors.

```
interaction.plot(humidity,environment,hours)
```



```
interaction.plot(environment,humidity,hours)
```



Task 3

We perform an ANOVA test to see the effect and interactions.

```
bread$humidity=as.factor(bread$humidity)
bread$environment=as.factor(bread$environment)
breadaov=lm(hours~humidity*environment,data=bread)
anova(breadaov)
```

We can see that environment and humidity have a main effect according to their p-values 2.46e-10 and 4.32e-06 respectively.

Looking at the p-value of humidity:environment = 3.71e-07, we see that there is interaction between the two factors.

In summary, this means that the hours it takes to get the bread to decay is influenced by the combination of this two factors and its combinations.

Task 4

```
contrasts(bread$humidity)=contr.sum
contrasts(bread$environment)=contr.sum
breadaov2=lm(hours~humidity*environment,data=bread)
```

```
summary(breadaov2)[[4]]
```

```
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  250.7       4.90   51.17 2.04e-15
## humidity1                     38.7       4.90    7.89 4.32e-06
## environment1                 149.3       6.93   21.55 5.81e-11
## environment2                 -64.7       6.93   -9.33 7.50e-07
## humidity1:environment1       -74.7       6.93  -10.78 1.59e-07
## humidity1:environment2        15.3       6.93    2.21 4.70e-02
```

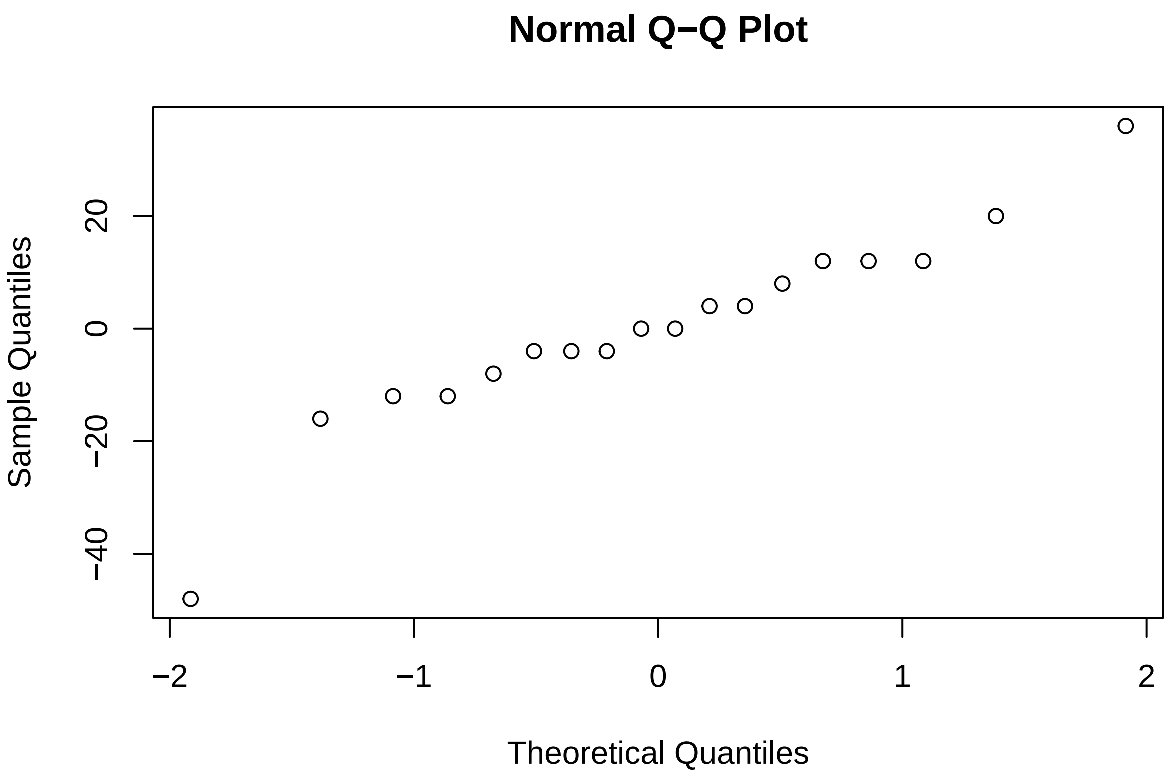
Here we can see that theenvironment factor affects the in a bigger way than humidity. We can see changes of [149.333] and [-64.667] in comparison with [38.667].

However, we don't think it's a good question, since we believe the core of the influence resides in the interaction between these two factors rather than only in one of them, even if it shows bigger changes.

Task 5

Finally, we will check the assumptions. First we perform a QQ-plot with the residuals for normality.

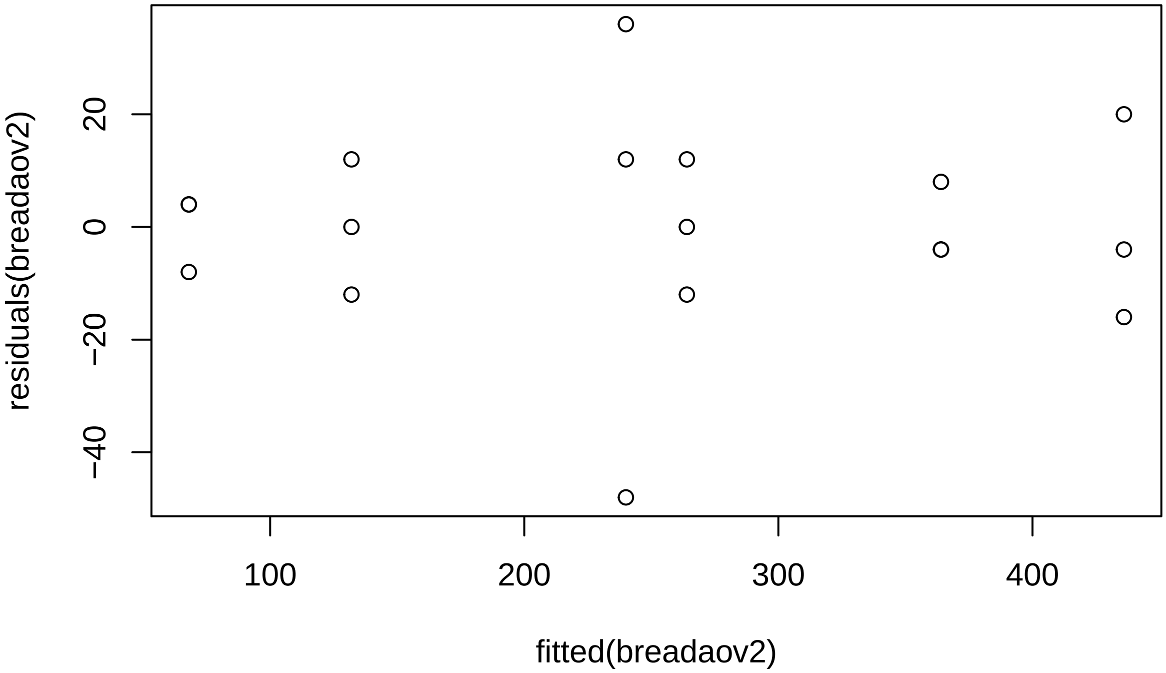
```
qqnorm(residuals(breadaov2))
```



The normality according to the QQ-plot could be doubtful because of the extreme values.

Next, we check the scatter plot of fitted values vs residuals for outliers.

```
plot(fitted(breadaov2),residuals(breadaov2))
```



In the plot we see that the values don't change systematically. We can see some outliers between 200 and 300.

Exercise 2

We load the data from the data source.

```
search=read.table("search.txt",header = TRUE);
attach(search)
```

Task 1

For the randomization task, we define the variables N (the dependent variable), I (the treatment) and B (the block). Then we perform the randomization.

```
N=1
I=3
B=5
```

```
for (i in 1:B){
  print(sample(1:(N*I)))
}
```

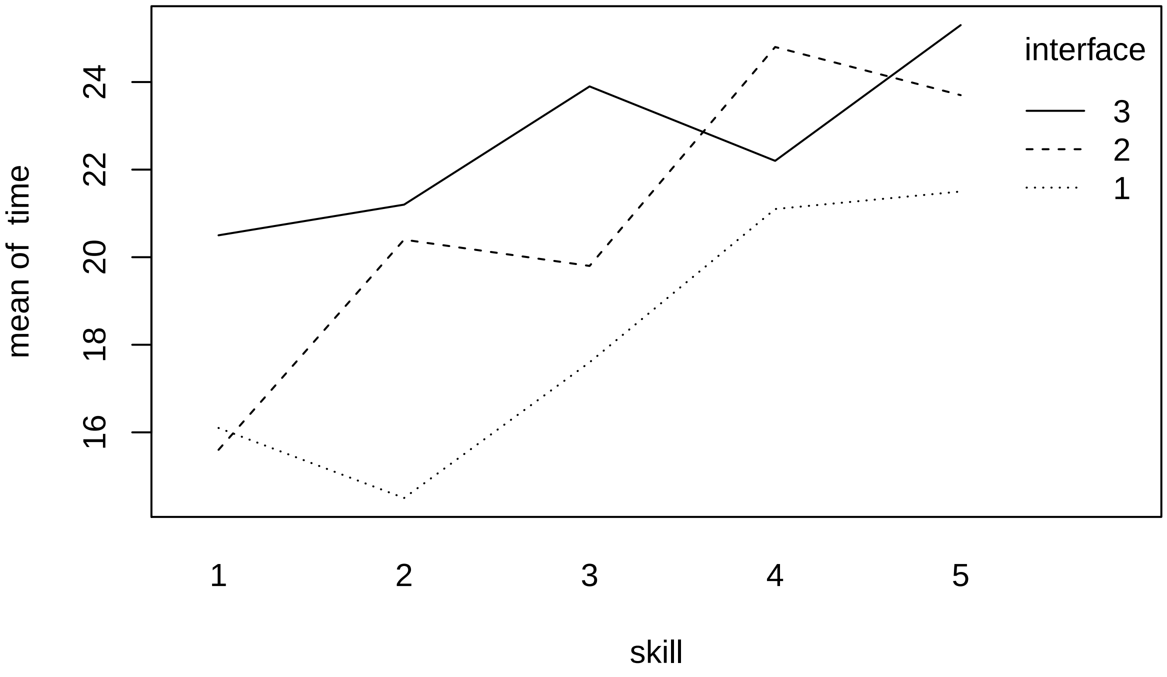
```
## [1] 1 2 3
## [1] 2 3 1
## [1] 3 1 2
## [1] 1 3 2
## [1] 3 1 2
```

The rows in the matrix represent the skill level of the students. Row 1 is level 1, row 2 is level 2, etc.

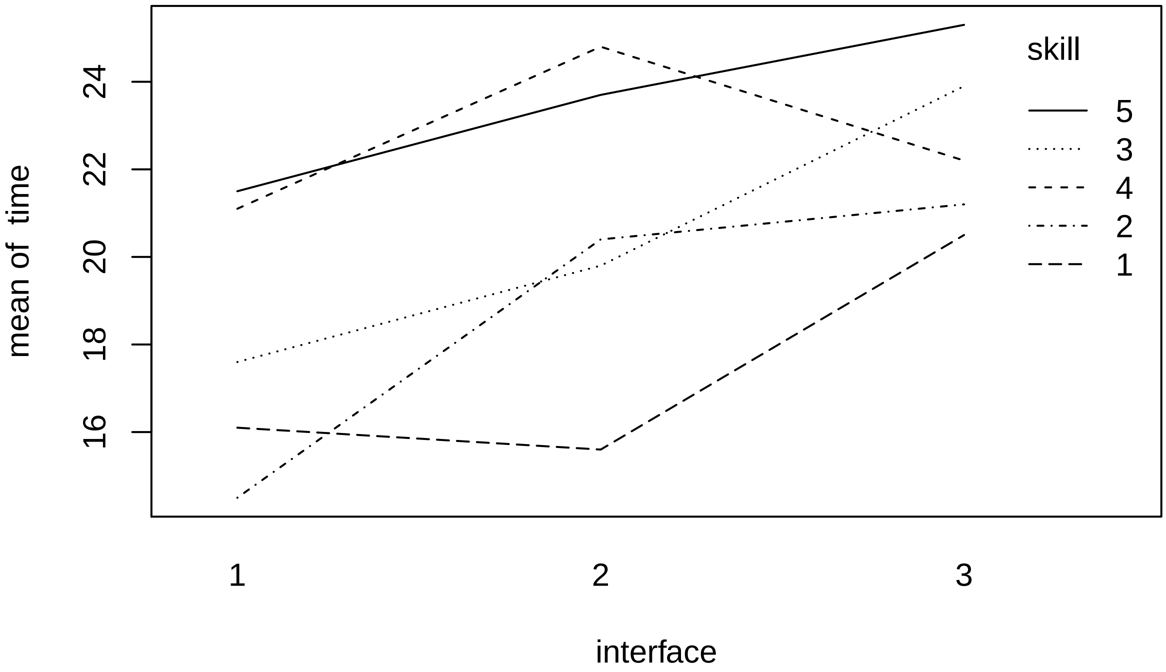
Each value of the row represent what interface has to be tested by one of the students of that level.

Task 2

```
interaction.plot(skill,interface,time)
```



```
interaction.plot(interface,skill,time)
```



The lines do not seem parallel. Therefore, we can say that there is interaction between interface and skill.

Task 3

We consider the variables as factors because they are numbers. Then, we perform a 2-way anova test.

```
search$skill = as.factor(search$skill)
search$interface = as.factor(search$interface)

searchaov = lm(time~interface+skill,data=search)
anova(searchaov)
```

We get a p-value for interface of 0.013. Which indicates that we can reject that the null hypothesis that the mean of the interfaces are the same.

Task 4

We perform the analysis.

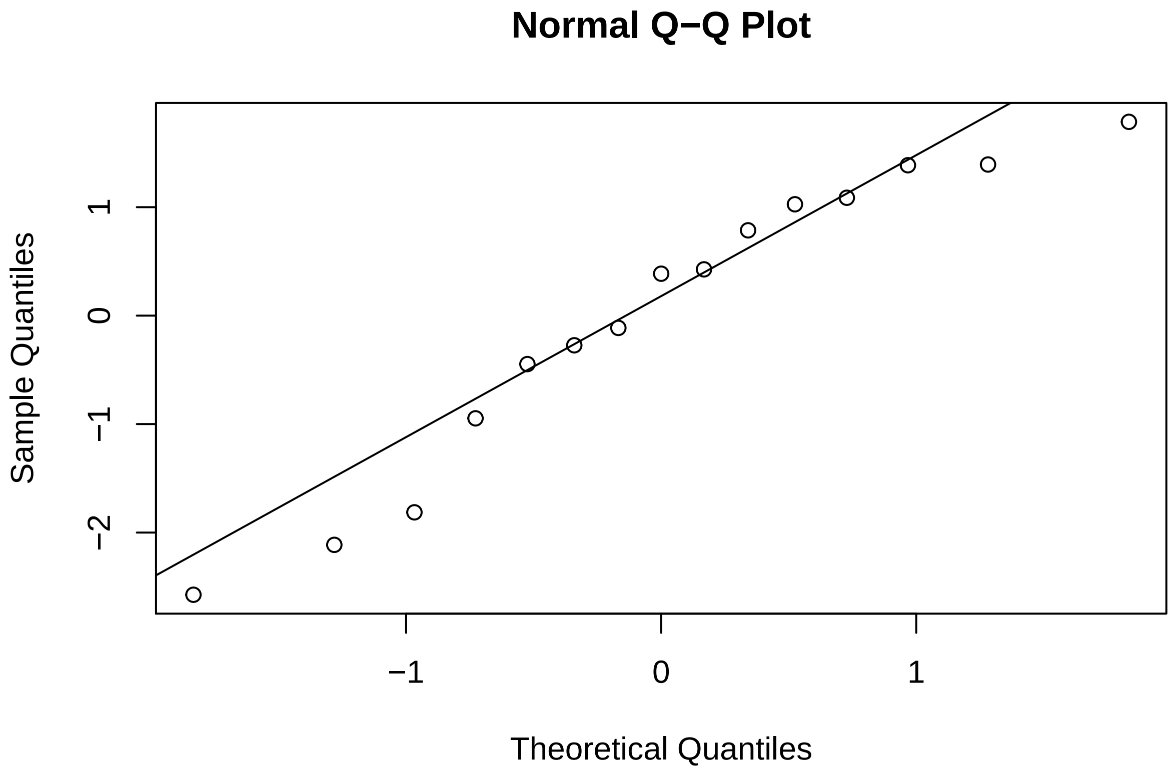
```r
searchaov2 = lm(time~interface+skill,data=search)

summary(searchaov2)
```

```
##
## Call:
## lm(formula = time ~ interface + skill, data = search)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.573 -0.697  0.387  1.057  1.787
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    15.01       1.23   12.24  1.8e-06 ***
## interface2      2.70       1.14    2.38   0.0447 *
## interface3      4.46       1.14    3.93   0.0044 **
## skill2          1.30       1.47    0.89   0.4012
## skill3          3.03       1.47    2.07   0.0724 .
## skill4          5.30       1.47    3.61   0.0068 **
## skill5          6.10       1.47    4.16   0.0032 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.8 on 8 degrees of freedom
## Multiple R-squared:  0.835,  Adjusted R-squared:  0.711
## F-statistic: 6.74 on 6 and 8 DF,  p-value: 0.0084
```

with the formula $\mathrm{Y}_{\mathrm{i,b,n}} = \mathrm{mu} + \mathrm{alpha}_{\mathrm{i}} + \mathrm{beta}_{\mathrm{b}} + \mathrm{error}$ we get $\mathrm{Y}_{4,3} = 15.01 + 5.3 + 4.46 = 24.77$, which is the estimate of a user with skill 4 who is using interface 3.

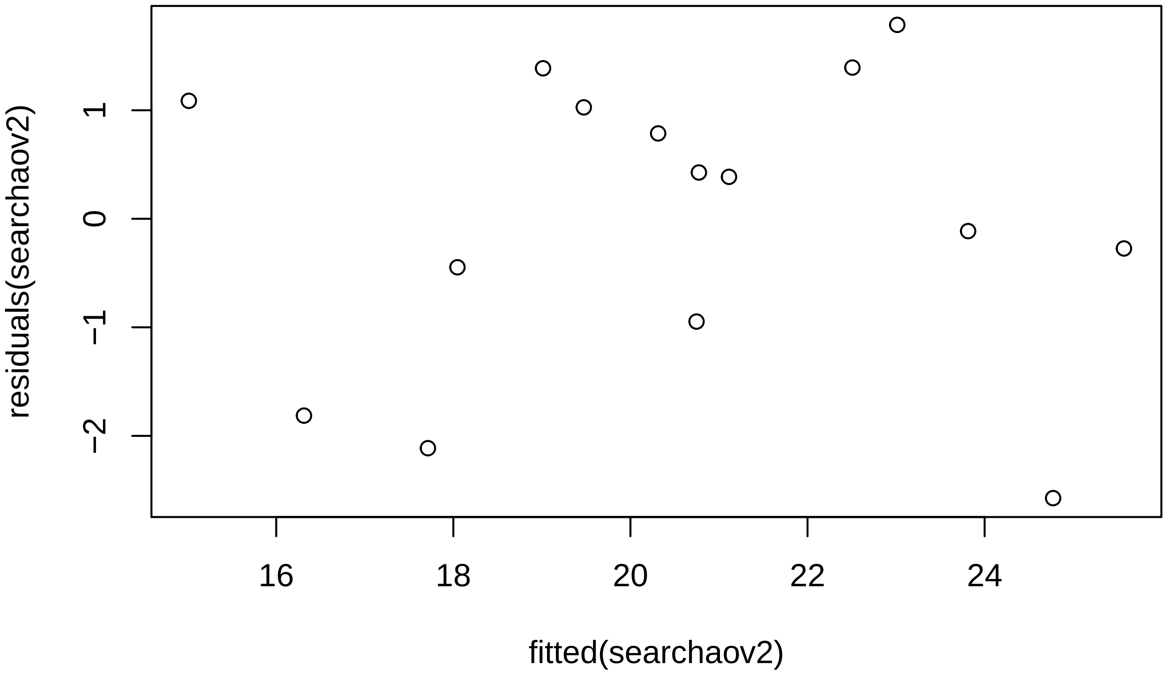
Task 5

```r
qqnorm(residuals(searchaov2))
qqline(residuals(searchaov2))
```



The QQ-plot seems a bit deviated in the extremes but it could be normal.

```r
plot(fitted(searchaov2),residuals(searchaov2))
```



We see that the residuals don't change systematically with the fitted values. So we can assume that the populations have equal variances.

Task 6

We perform the Friedman test.

```
friedman.test(search$time,search$interface,search$skill)
```

```
##
##  Friedman rank sum test
##
## data:  search$time, search$interface and search$skill
## Friedman chi-squared = 6, df = 2, p-value = 0.04
```

We reject H0 (Interface doesn't have an effect) so we can say that the interface makes an effect.

Task 7

we perform the one-way anova test.

```
search$interface = as.factor(search$interface)

searchaov = lm(time~interface,data=search)
```

```r
anova(searchaov)
```

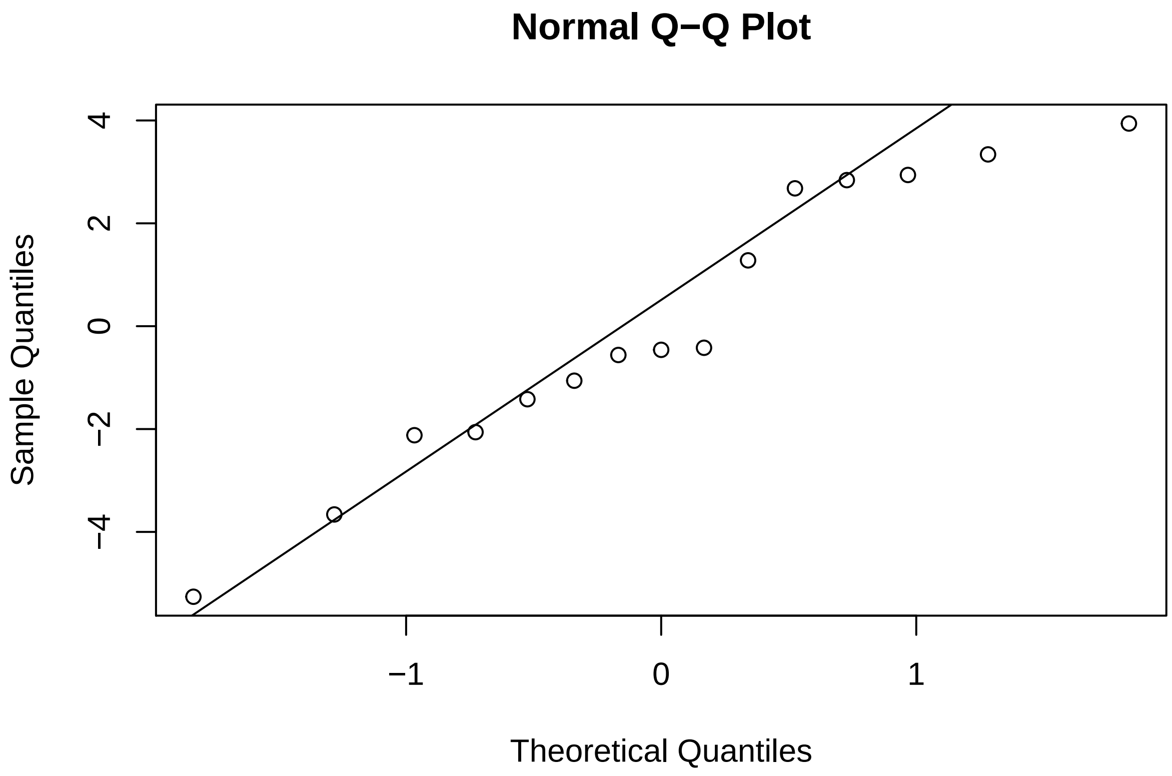
```
## Analysis of Variance Table
##
## Response: time
##           Df Sum Sq Mean Sq F value Pr(>F)
## interface  2   50.5   25.23    2.86  0.096 .
## Residuals 12  105.9    8.82
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We get p-value of 0.096 so we cannot reject that the means are the same for the different interfaces.

It is not useful because the variable skill should also be considered since according to the interaction plots there is interaction between skill and interface.

1-way-anova assumes that the data come from normal population and the variances are equal

```r
qqnorm(residuals(searchaov))
qqline(residuals(searchaov))
```



The population doesn't seem normal therefore the assumption is not met.

Exercise 3

We load the data from the data source.

```
cream = read.table("cream.txt", header=TRUE)
attach(cream)
```

Task 1

Then we perform the analysis in a three way experiments without interactions. After that, we do multi-factor anova test. In our case we have three factors.

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

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

```
## Analysis of Variance Table
##
## Response: acidity
##           Df Sum Sq Mean Sq F value  Pr(>F)
## starter    4   44.1   11.03   20.21 2.9e-05 ***
## batch      4   18.8    4.69    8.60  0.0016 **
## position   4    2.3    0.59    1.08  0.4112
## Residuals 12    6.6    0.55
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value for the factor “starter” is 2.904e-05, the p-value for factor “batch” is 0.002 and the p-value for the factor “position” is 0.41.

Therefore, we can conclude that we can’t reject the null hypothesis for position factor. So, position does not have any main effect. On the other hand, if we look into the p-values of both starter and batch then we can conclude that both of these factors have main effect. Since, we can reject our null hypothesis for both these factors.

Task 2

Now for this task we are taking the summary to produce the table of p-values.

```
creammult =glht(creamaov,linfct=mcp(starter="Tukey")) # For starter
summary(creammult)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
```

```
##
##
## Fit: lm(formula = acidity ~ starter + batch + position, data = cream)
##
## Linear Hypotheses:
##              Estimate Std. Error t value Pr(>|t|)
## 2 - 1 == 0    -0.150      0.467   -0.32     1.00
## 3 - 1 == 0    -0.980      0.467   -2.10     0.28
## 4 - 1 == 0     2.810      0.467    6.01   <0.001 ***
## 5 - 1 == 0    -0.484      0.467   -1.04     0.83
## 3 - 2 == 0    -0.830      0.467   -1.78     0.43
## 4 - 2 == 0     2.960      0.467    6.33   <0.001 ***
## 5 - 2 == 0    -0.334      0.467   -0.71     0.95
## 4 - 3 == 0     3.790      0.467    8.11   <0.001 ***
## 5 - 3 == 0     0.496      0.467    1.06     0.82
## 5 - 4 == 0    -3.294      0.467   -7.05   <0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

From the above table we can see that, the p-values that are less than 0.05 are: 4 -1 == 0, 4 - 2 == 0, 4 - 3 == 0, 5 - 4 == 0. Hence, he starter that leads to significant different acidity is number starter4.

Task 3

We are taking into consideration of the summary

```
creamaov = lm(acidity~starter+batch+position, data = cream)
summary(creamaov)
```

```
##
## Call:
## lm(formula = acidity ~ starter + batch + position, 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)     8.662      0.533   16.26  1.5e-09 ***
## starter2       -0.150      0.467   -0.32    0.754
## starter3       -0.980      0.467   -2.10    0.058 .
## starter4        2.810      0.467    6.01  6.1e-05 ***
## starter5       -0.484      0.467   -1.04    0.321
## batch2         -1.348      0.467   -2.88    0.014 *
## batch3          0.276      0.467    0.59    0.566
## batch4         1.368      0.467    2.93    0.013 *
## batch5         0.200      0.467    0.43    0.676
## position2     -0.618      0.467   -1.32    0.211
## position3     -0.038      0.467   -0.08    0.937
## position4     -0.764      0.467   -1.63    0.128
## position5     -0.264      0.467   -0.56    0.583
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.739 on 12 degrees of freedom
## Multiple R-squared:  0.909,  Adjusted R-squared:  0.818
## F-statistic: 9.96 on 12 and 12 DF,  p-value: 0.000178
```

From task 2, we get a p-value = 0.99 (for alpha2 - alpha1 = 0). Running the summary, we get the p-value = 0.75 (for alpha2 - alpha1 = 0). So, for simultaneous p-values we get a higher p-value compared to the p-value that we get running the summary for task 1. The reason behind this could be for simultaneous p-values the probabilty of getting accurate values is higher.

Task 4

We produce a table with confidence intervals.

```
confint(creammult)
```

```
##
##   Simultaneous Confidence Intervals
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = acidity ~ starter + batch + position, data = cream)
##
## Quantile = 3.19
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##            Estimate lwr    upr
## 2 - 1 == 0 -0.150   -1.640  1.340
## 3 - 1 == 0 -0.980   -2.470  0.510
## 4 - 1 == 0  2.810    1.320  4.300
## 5 - 1 == 0 -0.484   -1.974  1.006
## 3 - 2 == 0 -0.830   -2.320  0.660
## 4 - 2 == 0  2.960    1.470  4.450
## 5 - 2 == 0 -0.334   -1.824  1.156
## 4 - 3 == 0  3.790    2.300  5.280
## 5 - 3 == 0  0.496   -0.994  1.986
## 5 - 4 == 0 -3.294   -4.784 -1.804
```

The intervals that do not contain 0 are the following: 4 - 1 == 0 [1.3198,4.3002] 4 - 2 == 0 [1.4698 ,4.4502] 4 - 3 == 0 [2.2998,5.2802] 5 - 4 == 0 [-4.7842 -1.8038]

This indicates that the starter4 has main effect compared to the rest of other starters.

Exercise 4

First, we load the data from the data source.

```
cows = read.table("cow.txt", header = TRUE)
```

Task 1

```
cows$id=factor(cows$id)
cows$per=factor(cows$per)
cowslm=lm(milk~treatment+per+id,data=cows)
```

```
cowsanova=lm(milk~id+per+treatment,data=cows)
anova(cowsanova)
```

```
## Analysis of Variance Table
##
## Response: milk
##           Df Sum Sq Mean Sq F value  Pr(>F)
## id         8   2467   308.4  124.48 7.5e-07 ***
## per        1     25    24.5    9.89   0.016 *
## treatment  1      1     1.2    0.47   0.517
## Residuals  7     17     2.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we can see after running anova we get a p-value = 0.52, which is greater than 0.05. Therefore, we cannot reject the hypothesis that treatment doesn't have any main effect on milk production.

Task 2

```
milkdif =glht(cowslm,linfct=mcp(treatment="Tukey"))
```

We can see that the estimate of B-A is [-0.51]. In addition to this, we see thatthe p-value is [0.517].

With this, we can't reject that the difference in milk production with treatment A is equal to the one using treatment B.

Task 3

Checking influence in milk production

```
cowslmer=lmer(milk~treatment+order+per+(1|id),data=cows,REML=FALSE)
cowslmer1=lmer(milk~order+per+(1|id),data=cows,REML=FALSE)
anova(cowslmer1,cowslmer)
```

As we can see in the p-values [0.446] of the tratment, we can see that it is higher than 0.05. Thus, we cannot reject the hypothesis that treatment doesn't have any main effect on milk production.

Estimating the difference

```
milkdiflmer =glht(cowslmer,linfct=mcp(treatment="Tukey"))
```

We can see that the estimate of B-A is [-0.51]. In addition to this, we see that the p-value is [0.439].

With this, we can't reject that the difference in milk production with treatment A is equal to the one using treatment B.

Task 4

The following command performs a t-test of the milk production using treatment a and milk production using treatment B. The test will help us say if the difference between these is 0 or not.

```
attach(cows)
t.test(milk[treatment=="A"],milk[treatment=="B"],paired=TRUE)
```

The t-test gives back the p-value [0.828]. With this, we can't reject that the difference is = to 0.

This test does agree with the conclussion in task 1, where we concluded that the feeding treatment didn't have main influence in the milk production. Which is similar to saying that the difference of the means of milk production between both treatments is 0. Thus, we believe the test is valid to come to the same result

Exercise 5

We load the data from the data source.

```
nauseadata = read.table("nauseatable.txt", header = TRUE)
attach(nauseadata)
```

Task 1

We make a vecor of 0's and 1's for the incidence of no nausea and nausea respectively.

```
nonausea_detected = nauseadata[, 1]

nausea_detected = nauseadata[, 2]

nausea = as.vector(rep(1, each = nausea_detected[1]+
                            nausea_detected[2]+
                         nausea_detected[3]))

nonausea = as.vector(rep(0, each = nonausea_detected[1]+
                           nonausea_detected[2]+
                           nonausea_detected[3]))

nausea_vector = c(nonausea, nausea)
```

Then we define a function to label the medicines based on the incidence of nausea and no nausea.

```
label_medicin <- function(label, count){

   med_label <- as.vector(rep(label, count))

   return(med_label)

}
```

Now we make a vector of medicine with different labels.

```
# Total count of medicines for No nausea
C_nonausea = 100
P1_nonausea = 32
P2_nonausea = 48

# Total count of medicines for Nausea
C_nausea =  52
P1_nausea = 35
P2_nausea = 37


medicin_vector = c(
  label_medicin("C", C_nonausea),
  label_medicin("P1", P1_nonausea),
  label_medicin("P2", P2_nonausea),
  label_medicin("C", C_nausea),
  label_medicin("P1", P1_nausea),
  label_medicin("P2", P2_nausea))
```

After that we set-up our data frame that includes 304 rows in total and two cloumns for nausea and medicine.

```
nausea.frame = data.frame(nausea_vector, medicin_vector)
```

We assigned labels "C" for Chlorpromazine, "P1" for Pentobarbital(100mg) and "P2" for Pentobarbital(150mg).

We could have assigned same label for both doses of Pentobarbital. But we want to be specific so we used different labels for each one.

Task 2

To study the outcome of xtabs we execute the following function:

```
xtabs(~medicin_vector+nausea_vector, data = nausea.frame)
```

```
##               nausea_vector
## medicin_vector   0   1
##             C  100  52
##             P1  32  35
##             P2  48  37
```

In this case R does not give the warning, since the approximation is reliable for nausea_vector and medicin_vector in nausea.frame

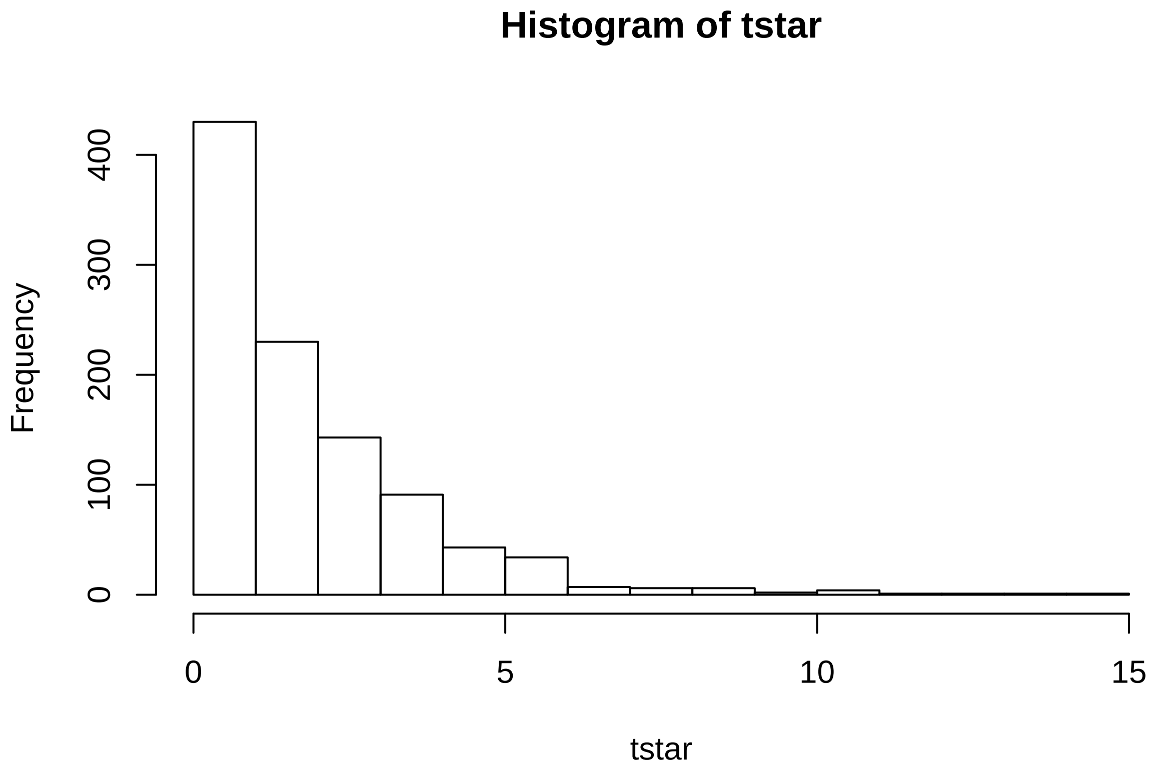
And if we study the outcome, we see that with xtabs we get a contingency table from the medicine and nausea factors.

We can see that there are more people suffering from nausea with the medicine Chlorpromazine than with Pentobarbital.

Task 3

We perform the permutation test.

```
B = 1000
tstar =  numeric(B)
for (b in 1:B) {
 medicinstar = sample(medicin_vector)
 tstar[b] = chisq.test(xtabs(~medicinstar+nausea_vector, data = nausea.frame))[[1]]
}

myt = chisq.test(xtabs(~medicin_vector+nausea_vector, data = nausea.frame))[[1]]

hist(tstar)
```



```
pl = sum(tstar<myt)/B
pr = sum(tstar>myt)/B

pmin = min(pl,pr)
pvalue = 2*pmin
```

Findings: We get a p-value = 0.052. Therefore, we can not reject that the two medicines have a significant difference. So, we can conclude that, both of them work equally well for nausea.

Task 4

In order to perform the chisq test we run the following code block:

```
pvalue_chisq = chisq.test(xtabs(~medicin_vector+nausea_vector, data = nausea.frame))[[3]]

pvalue_tstar=2*pmin
```

If we look into the outcome from above code, we get a p-value = 0.036 for chisq test. While on the other hand, by executing the permutation test we get a p-value = 0.052. Observing these two p-values we can say that they are different.

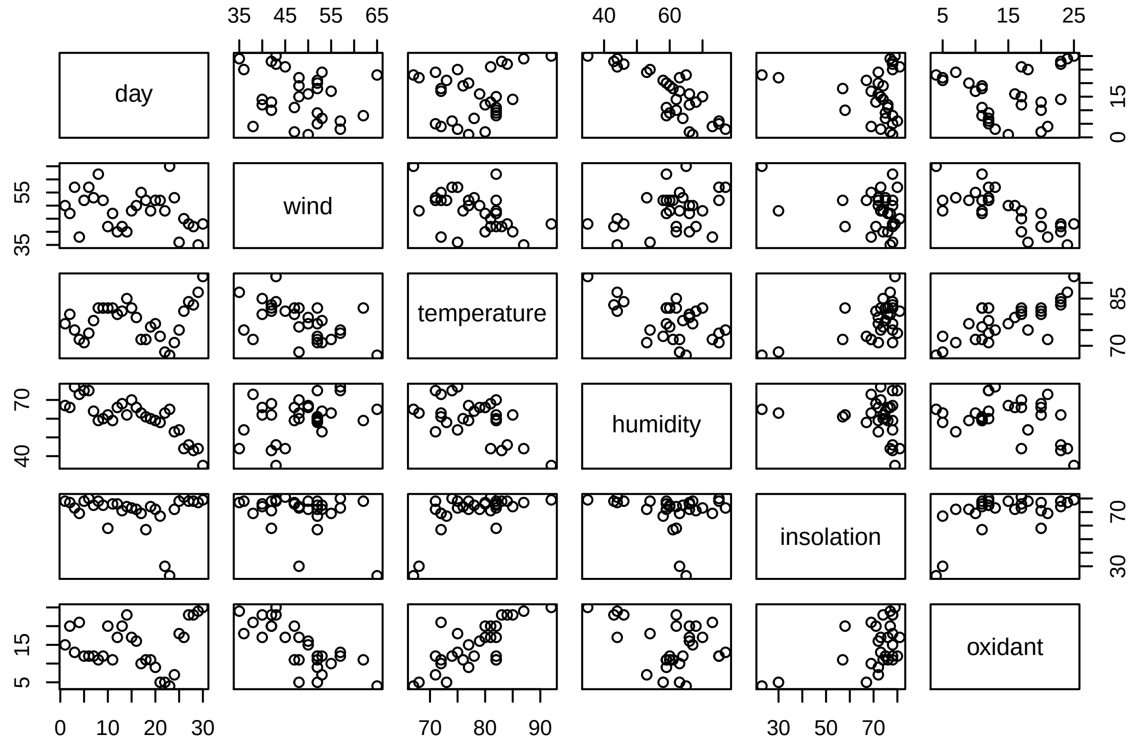
Exercise 6

First we load the data from the data source.

```
pollution = read.table("airpollution.txt",header=TRUE)
```

Task 1

```
pairs(pollution)
```



Looking at the scatter plots, there seems to be a correlation between wind and humidity, temperature and oxidant. Therefore we will check the correlation.

```
round(cor(pollution[,2:6]),2)
```

```
##              wind temperature humidity insolation oxidant
## wind         1.00       -0.50     0.37      -0.32   -0.77
## temperature -0.50        1.00    -0.54       0.57    0.76
## humidity     0.37       -0.54     1.00      -0.18   -0.35
## insolation  -0.32        0.57    -0.18       1.00    0.51
## oxidant     -0.77        0.76    -0.35       0.51    1.00
```

Looking at the table, we can see that our assumptions are not true, there seems to be no correlation between these variables.

Task 2

```
# Add Code
```

First we perform the test on each of the explanatory variables to get the best.

```
poplm = lm(oxidant~wind,data=pollution)
```

Multiple R-squared: [0.586]

```
poplm = lm(oxidant~temperature,data=pollution)
```

Multiple R-squared: [0.576]

```
poplm = lm(oxidant~humidity,data=pollution)
```

Multiple R-squared: [0.124]

```
poplm = lm(oxidant~insolation,data=pollution)
```

Multiple R-squared: [0.255]

The variable that gives higher R-squared is wind. Therefore, we add it to our model.

```
poplm = lm(oxidant~wind+temperature,data=pollution)
```

Multiple R-squared: [0.777]

```
poplm = lm(oxidant~wind+humidity,data=pollution)
```

Multiple R-squared: [0.591]

```
poplm = lm(oxidant~wind+insolation,data=pollution)
```

Multiple R-squared: [0.661]

The one with higer value is temperature. So we also add it to our model and continue looking for variables.

```
poplm = lm(oxidant~wind+temperature+humidity,data=pollution)
```

Multiple R-squared: [0.796]

```
poplm = lm(oxidant~wind+temperature+insolation,data=pollution)
```

Multiple R-squared: [0.782]

Now we can see that none of these are relevant for the linear regression.

Our final linear regression model using step-up is: oxidant = -5.0 -0.43*wind + 0.52*temperature

Task 3

```
poplm = lm(oxidant~wind+temperature+humidity+insolation,data=pollution)
summary(poplm)[[4]]
```

The one with highest p-value is insolation. Since it is bigger than 0.05, we delete insolation from the model.

```
poplm = lm(oxidant~wind+temperature+humidity,data=pollution)
summary(poplm)[[4]]
```

The one with highest p-value is humidity Since it is bigger than 0.05, we delete insolation from the model.

```
poplm = lm(oxidant~wind+temperature,data=pollution)
summary(poplm)[[4]]
```

There's no variable with a p-value bigger than 0.05. Thus, all variables are significant for our model.

The final model using step-down is: oxidant = -5.0 -0.43*wind + 0.52*temperature

Task 4

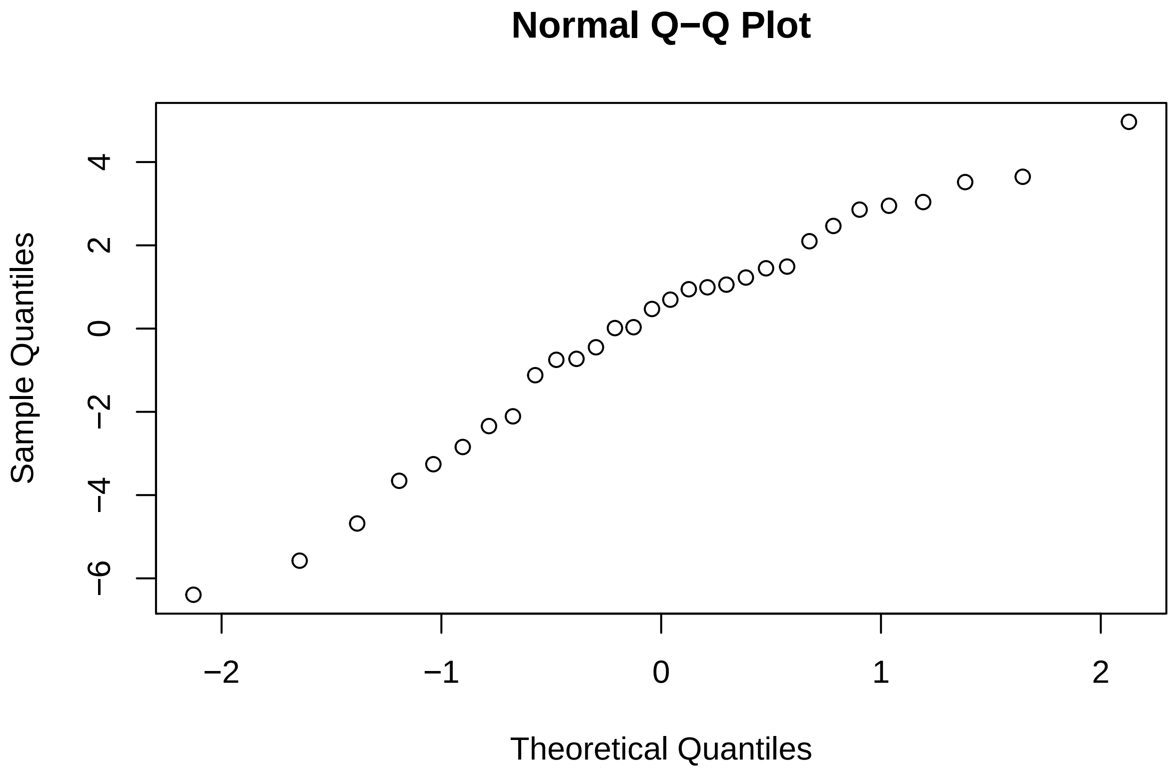
Since both methods lead us to the same model, we don't need to compare them to choose.

Estimates: wind = 0.43, temperature = 0.52, humidity = 0, insolation = 0;

Task 5

To check the normality we're gonna use a QQ-plot.

```
qqnorm(residuals(poplm))
```



According to the qqplot the residuals seem normal.

Exercise 7

First, we load the data from the data source.

```
expcrime =  read.table("expensescrime.txt", header=TRUE)
```

The next step is choosing our model. For this purpose, we will conduct both "Step-Up" and "Step-Down" methods and compare the resulting models.

Step-Up Method

With this method, we must choose one by one the most significant variable for our model.For this, we must check their Multiple R-squared value and choose the highest one.

```
#Variable bad
crimelm=lm(expend~bad,data=expcrime)
summary(crimelm)[[4]]
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    126.7     114.86     1.1 2.75e-01
## bad             13.3       1.26    10.6 2.80e-14
```

Multiple R-squared: [0.696]

```r
#Variable crime
crimelm=lm(expend~crime,data=expcrime)
summary(crimelm)[[4]]
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -531.039    577.166   -0.92   0.3620
## crime          0.287      0.116    2.48   0.0165
```

Multiple R-squared: [0.112]

```r
#Variable lawyers
crimelm=lm(expend~lawyers,data=expcrime)
summary(crimelm)[[4]]
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -59.6120    53.7994   -1.11 2.73e-01
## lawyers       0.0704     0.0026   27.06 4.02e-31
```

Multiple R-squared: [0.937]

```r
#Variable pop
crimelm=lm(expend~pop,data=expcrime)
summary(crimelm)[[4]]
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -188.767   69.67628   -2.71 9.27e-03
## pop            0.217    0.00992   21.90 5.83e-27
```

Multiple R-squared: [0.907]

```r
#Variable employ
crimelmsu=lm(expend~employ,data=expcrime)
summary(crimelm)[[4]]
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -188.767   69.67628   -2.71 9.27e-03
## pop            0.217    0.00992   21.90 5.83e-27
```

Multiple R-squared: [0.954]

Since the variable employ is the one with the biggest Multiple R-squared value, we add it to our model and we continue to the next iteration. Here, we repeat the process but we add the selected variable to the formula.

```r
#Variable bad
crimelm=lm(expend~employ+bad,data=expcrime)
summary(crimelm)[[4]]
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -116.4498   46.96559   -2.48 1.67e-02
## employ         0.0497    0.00299   16.63 1.48e-21
## bad           -1.0898    0.99481   -1.10 2.79e-01
```

Multiple R-squared: [0.955]

```
#Variable crime
crimelm=lm(expend~employ+crime,data=expcrime)
summary(crimelm)[[4]]
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -248.3631   1.32e+02   -1.89 6.50e-02
## employ         0.0463   1.54e-03   30.01 9.37e-33
## crime          0.0296   2.76e-02    1.07 2.89e-01
```

Multiple R-squared: [0.955]

```
#Variable pop
crimelm=lm(expend~employ+pop,data=expcrime)
summary(crimelm)[[4]]
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -126.5921   50.21637  -2.521 1.51e-02
## employ         0.0433    0.00616   7.026 6.72e-09
## pop            0.0174    0.02930   0.594 5.55e-01
```

Multiple R-squared: [0.954]

```
#Variable lawyers
crimelm=lm(expend~employ+lawyers,data=expcrime)
summary(crimelm)[[4]]
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -110.6588   42.56735   -2.60 1.24e-02
## employ         0.0297    0.00511    5.81 4.89e-07
## lawyers        0.0269    0.00776    3.46 1.13e-03
```

Multiple R-squared: [0.963]

The variable lawyers is the one with a higher Multiple R-squared value. However, if we look at the different variables, adding any of these would yield insignificant explanatory variables. The change would be +0.001 at best compared with the last model. Therefore, we discard these variables and we should stop at the previous step to finalize our model.

The resulting model after the step-up method is:

expend = -1.167e+02 + 4.681e-02*employ + error

Step-Down Method

With this method, we start with all of the possible variables in our model. Then, we choose the one that gives the highest p-value. If this p-value is bigger than 0.05, we will discard the variable and repeat the process without it.

```
crimelm=lm(expend~employ+bad+crime+lawyers+pop,data=expcrime)
summary(crimelm)[[4]]
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -299.1341   1.40e+02   -2.14  0.03817
## employ         0.0230   7.46e-03    3.08  0.00354
## bad           -2.8319   1.24e+00   -2.28  0.02719
## crime          0.0324   2.81e-02    1.15  0.25534
## lawyers        0.0232   8.04e-03    2.89  0.00592
## pop            0.0779   3.51e-02    2.22  0.03184
```

The variable crime has the highest p-value with a p-value of [0.255]. Since this p-value is bigger than 0.05, we discard it for our model and continue to the next iteration.

```
crimelm=lm(expend~employ+bad+lawyers+pop,data=expcrime)
summary(crimelm)[[4]]
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -146.4239   45.41009   -3.22  0.00232
## employ         0.0228    0.00749    3.05  0.00380
## bad           -2.2407    1.13321   -1.98  0.05402
## lawyers        0.0265    0.00757    3.50  0.00106
## pop            0.0637    0.03304    1.93  0.06012
```

The variable pop has the highest p-value with a p-value of [0.06]. Since this p-value is bigger than 0.05, we discard it for our model and continue to the next iteration.

```
crimelm=lm(expend~employ+bad+lawyers,data=expcrime)
summary(crimelm)[[4]]
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -110.5809   42.60733  -2.595 1.26e-02
## employ         0.0323    0.00580   5.569 1.20e-06
## bad           -0.8627    0.90425  -0.954 3.45e-01
## lawyers        0.0263    0.00779   3.379 1.47e-03
```

The variable bad has the highest p-value with a p-value of [0.345]. Since the p-value is bigger than 0.05, we discard it for our model and continue to the next iteration.

```
crimelmsd=lm(expend~employ+lawyers,data=expcrime)
summary(crimelm)[[4]]
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -110.5809   42.60733  -2.595 1.26e-02
## employ         0.0323    0.00580   5.569 1.20e-06
## bad           -0.8627    0.90425  -0.954 3.45e-01
## lawyers        0.0263    0.00779   3.379 1.47e-03
```

As we can see, all the p-values are smaller than 0.05, thu meaning that all the variables are significant for our model.

The resulting model of the step-down method is:

expend = -1.107e+02 + 2.971e-02*employ + 2.686e-02*lawyers + error

Comparing the models

R values

Comparing the R-values of both models, we can see that the R-value of the step-down model is higher than the one of the step-up model. In this case a higher value is better.

Step-up model: Multiple R-squared = [0.954]

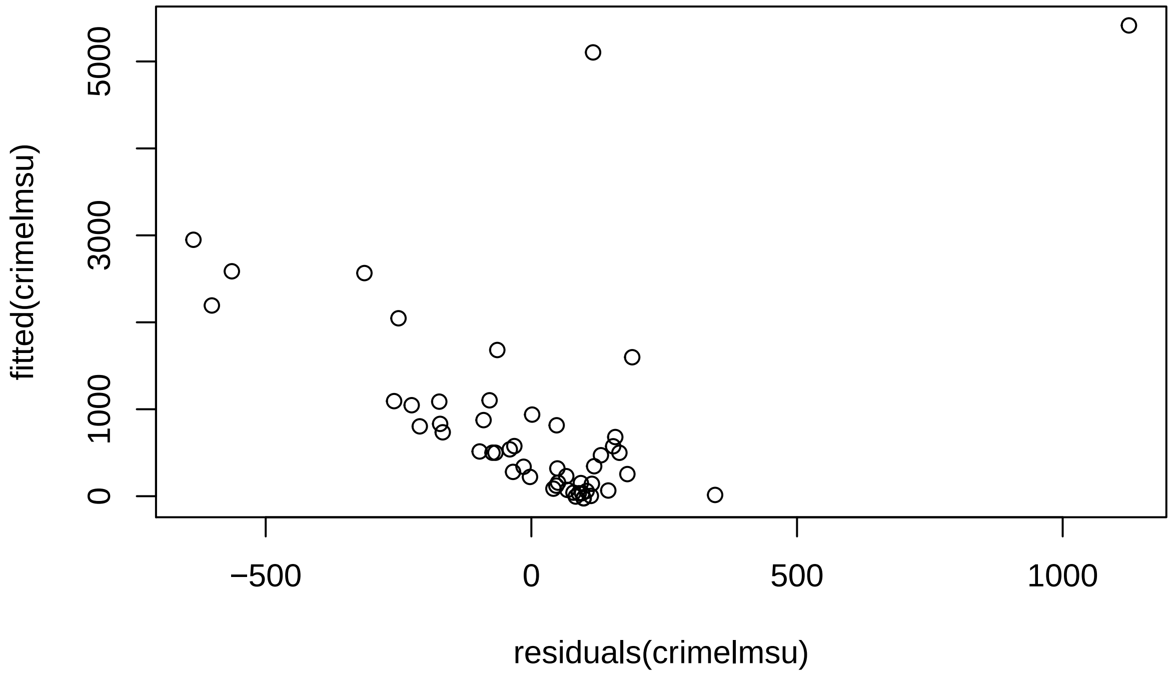
Step-down model: Multiple R-squared = [0.963]

Plots: Fitted values vs Residuals

Comparing the Fitted vs Residuals plots of both models, we look for specific structures.

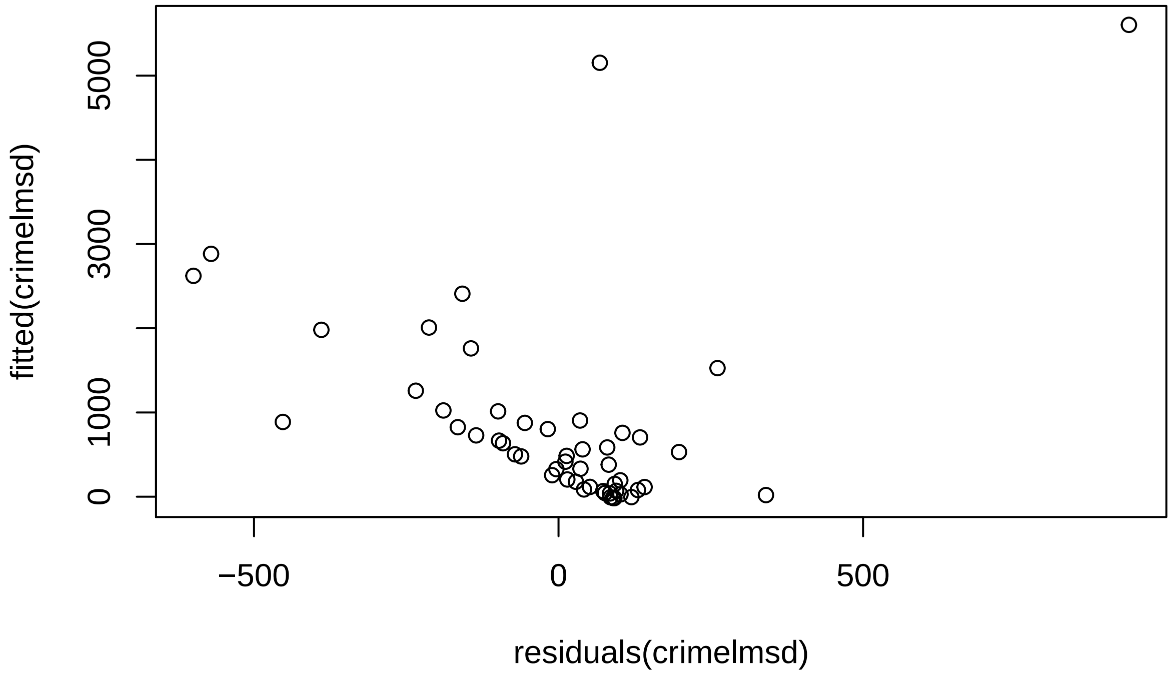
Step-up model plot:

```r
plot(residuals(crimelmsu), fitted(crimelmsu))
```



Step-down model plot:

```r
plot(residuals(crimelmsd), fitted(crimelmsd))
```



As we can see, none of them follow any specific structure so we can conclude both models are equally fit in this aspect.

Explanatory variables

In this case, we compare the number of explanatory variables per model. Looking at the models, we see that the step-up model has 1 explanatory variable (employ), whereas the step-down model has 2 (employ+lawyers). In this case, less variables are better.

Character of the explanatory variables

It is important that we check whether the explanatory variables are difficult to interprete. In this case, both variables are numerical. The variable "employ" makes reference to the number of persons employed in the state. On the other hand, the variable "lawyers" is the number of lawyers in the state.

Our choice

We've seen that both models don't differ that much one from another. However, we decided to choose the model provided by the step-up method. We chose this model since it has one less explanatory variable. Although the R value is higher for the step-down model, it only differs by 0.09, which in our opinion is not a big loss and it allows us to use the model with one less variable.

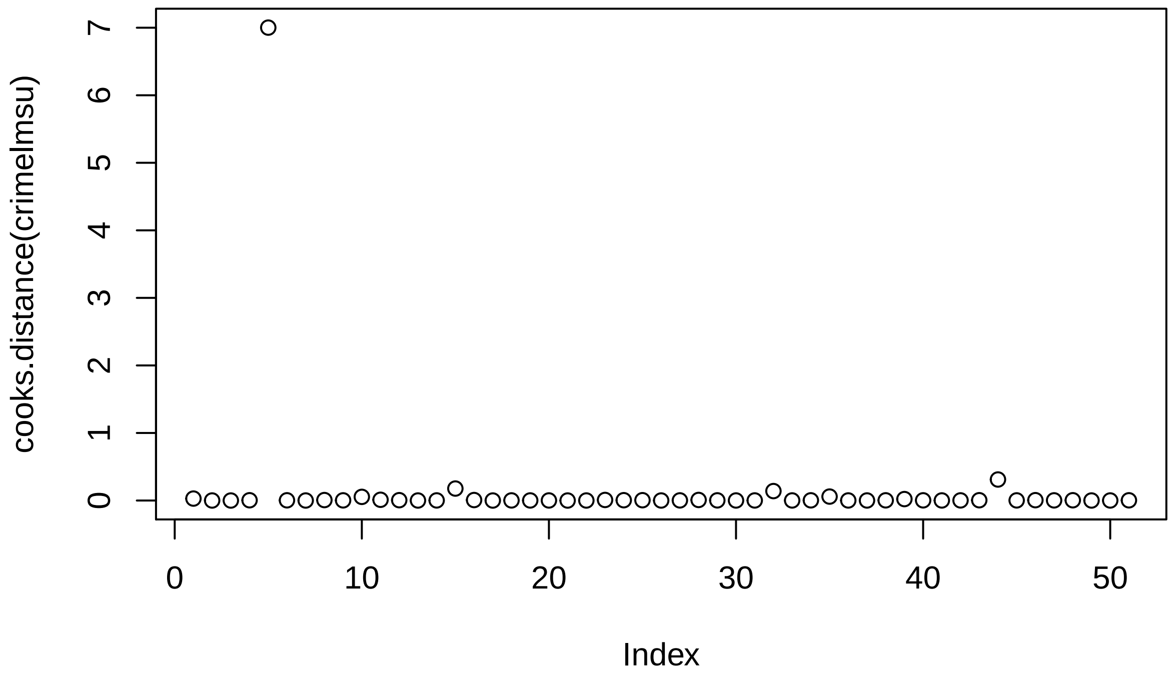
Task a

In order to investigate the influence points, we calculate the cooks distance and make a plot to spot the outliers.

```
round(cooks.distance(crimelmsu), 2)
```

Plot:

```
plot(cooks.distance(crimelmsu))
```

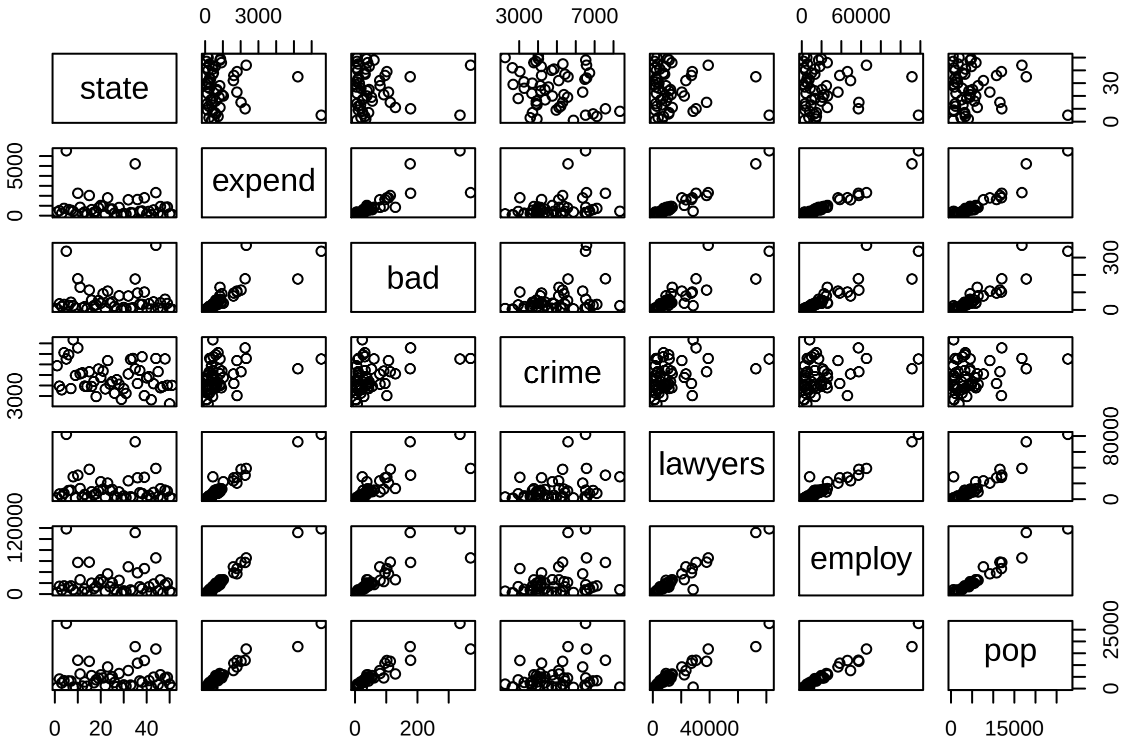


Here we clearly have encountered an influence point: the Cook's distance is [7] for the potential point.

Task b

To investigate collinearity problems, we first have to check the pairs with the scatter plot.

```
pairs(expcrime)
```



Here we can see that the variables “expend” and “lawyers” look somewhat collinear. We will now check the linear correlation between both.

```
round(cor(expcrime[,5:6]),2)
```

```
##          lawyers employ
## lawyers     1.00   0.97
## employ      0.97   1.00
```

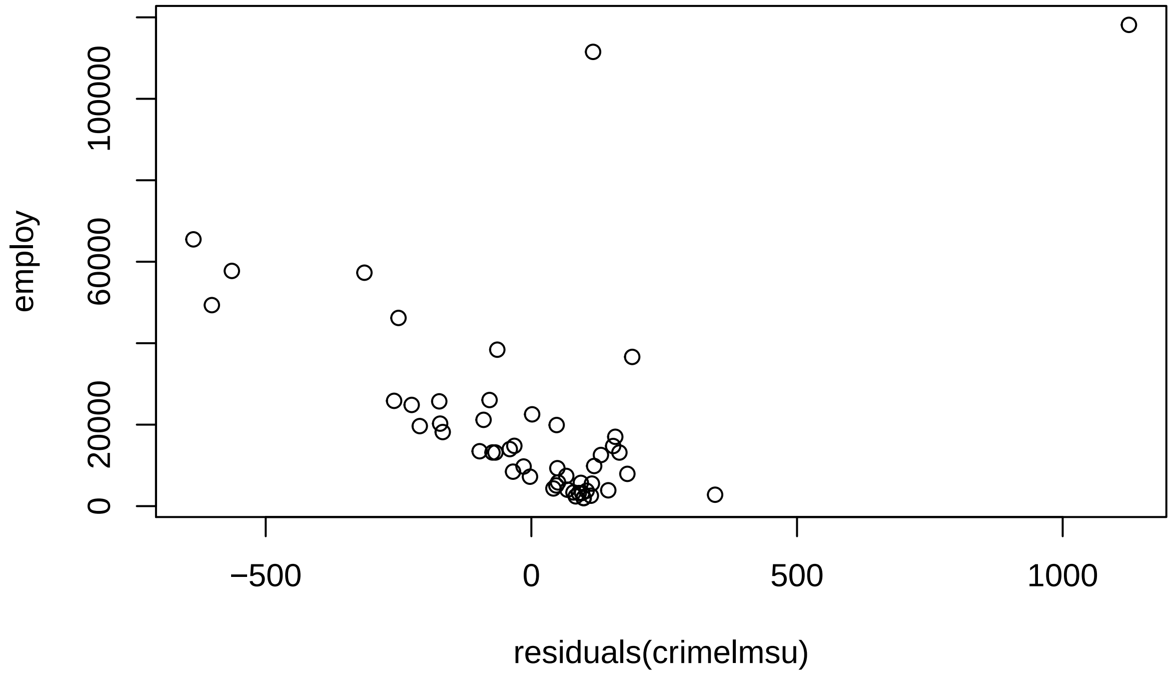
As suspected, both variables are collinear with [0.97]. This agrees with our model since we can’t have 2 collinear variables in the same model.

Task c

As a final task, we will investigate the residuals. For this we will use several plots.

First, a scatter plot of the residuals against the variable in the model to look at the pattern and spread.

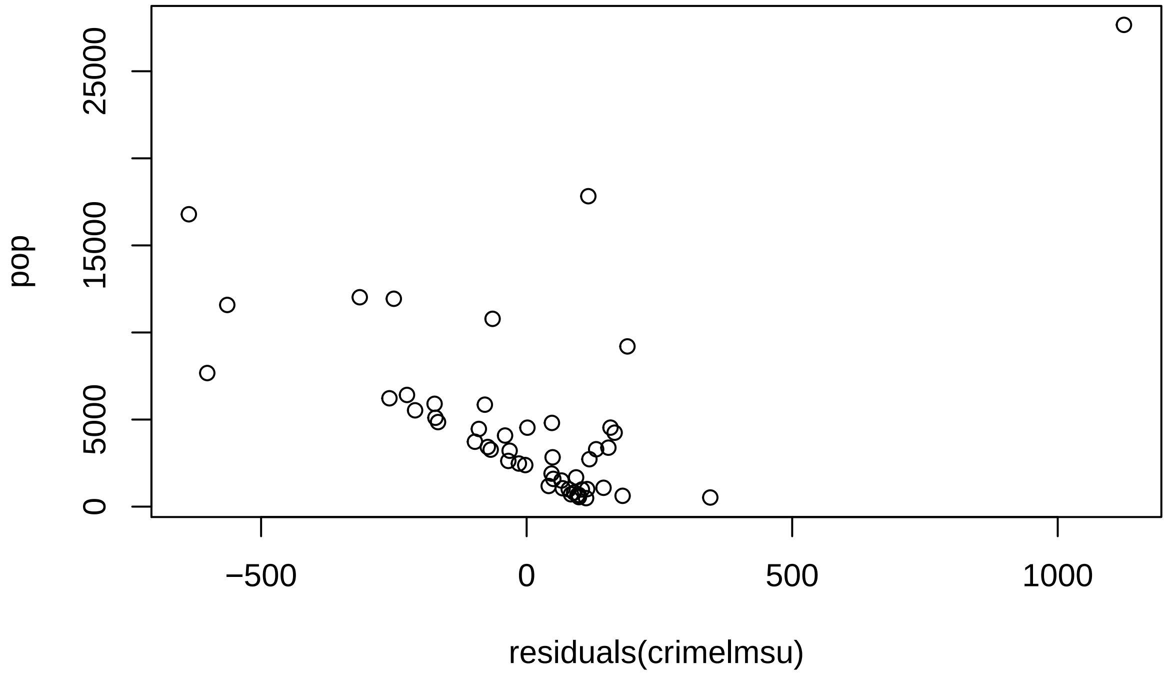
```
attach(expcrime)
plot(residuals(crimelmsu),employ)
```



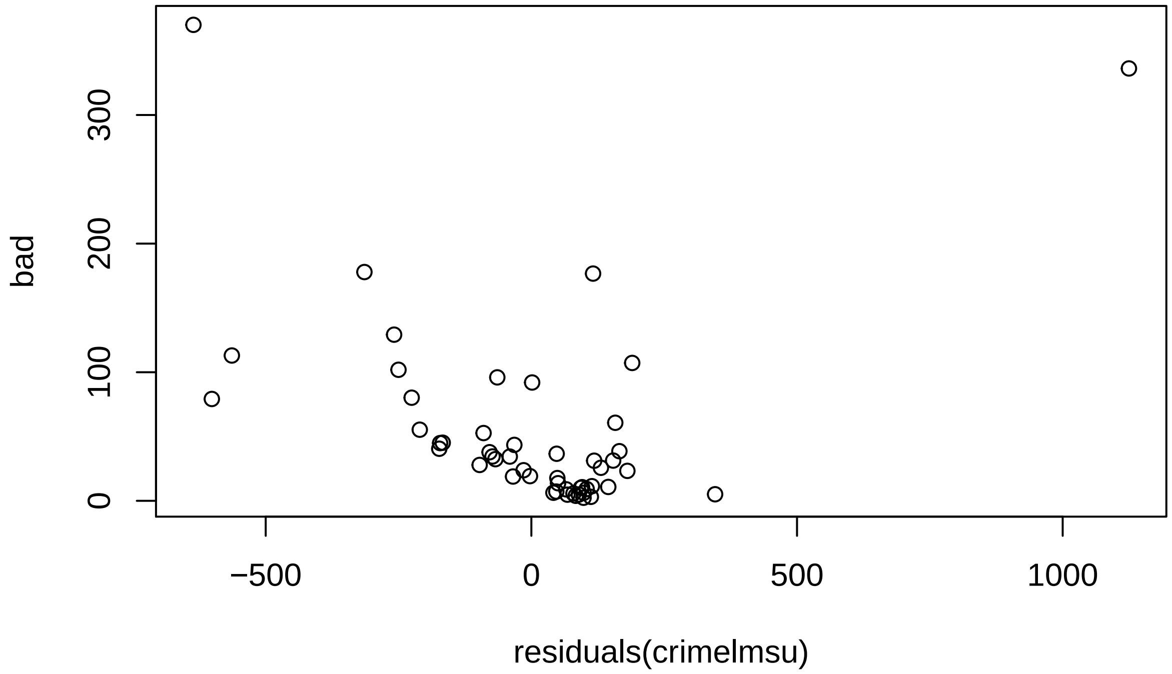
They don't seem to follow any kind of curved structure, which would mean that there's no linear relation between them.

In continuation, another scatter plot but this time against the variables that are not in the model.

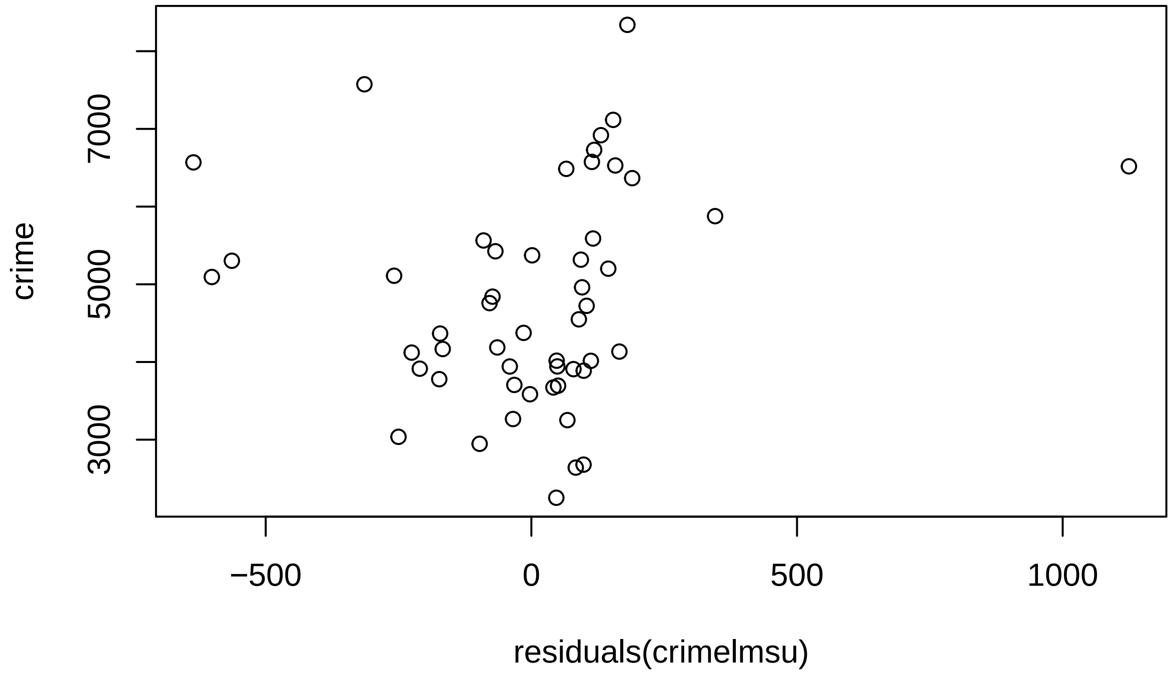
```r
plot(residuals(crimelmsu),pop)
```



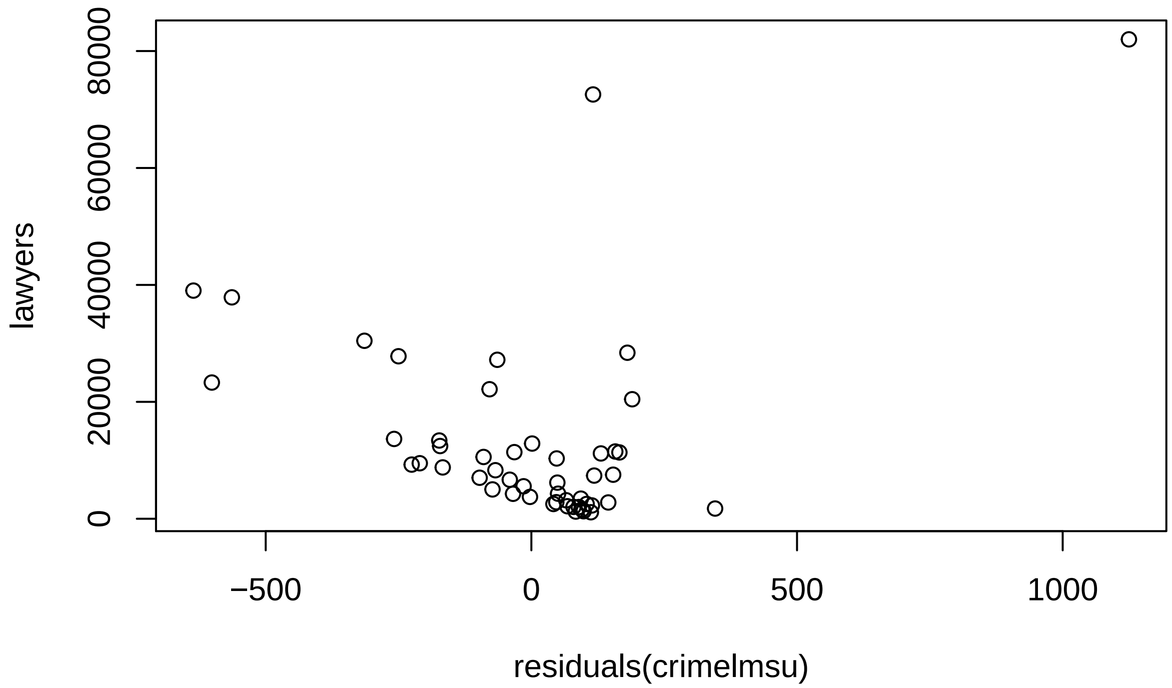
```
plot(residuals(crimelmsu),bad)
```



```r
plot(residuals(crimelmsu),crime)
```



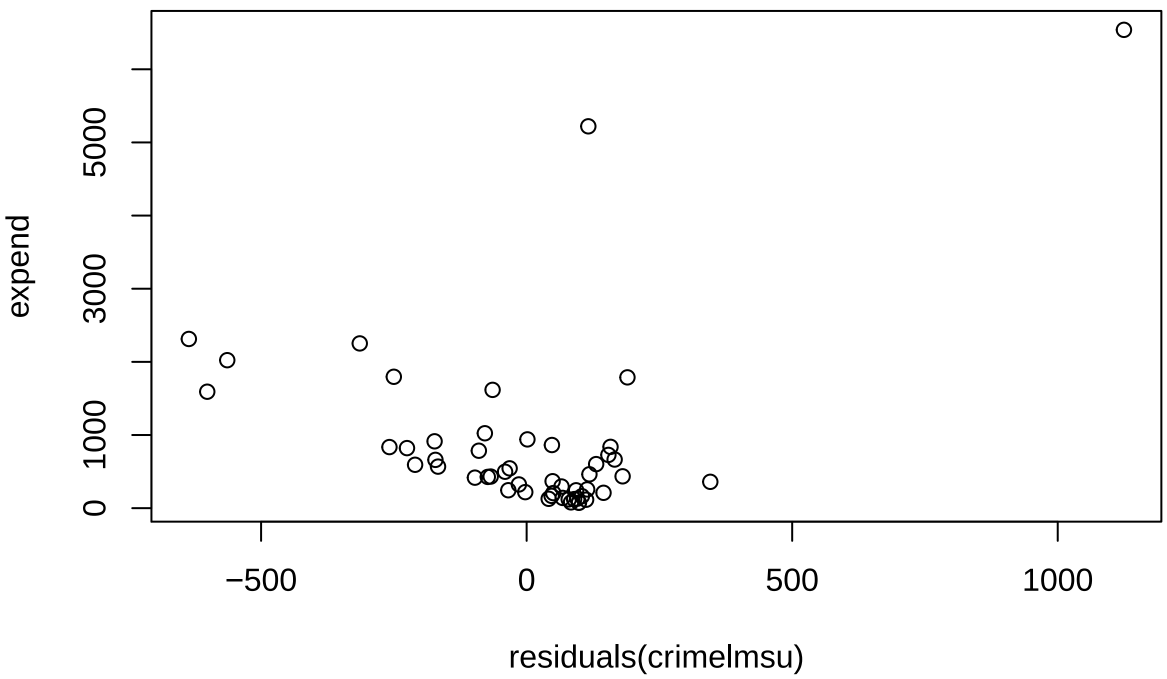
```r
plot(residuals(crimelmsu),lawyers)
```



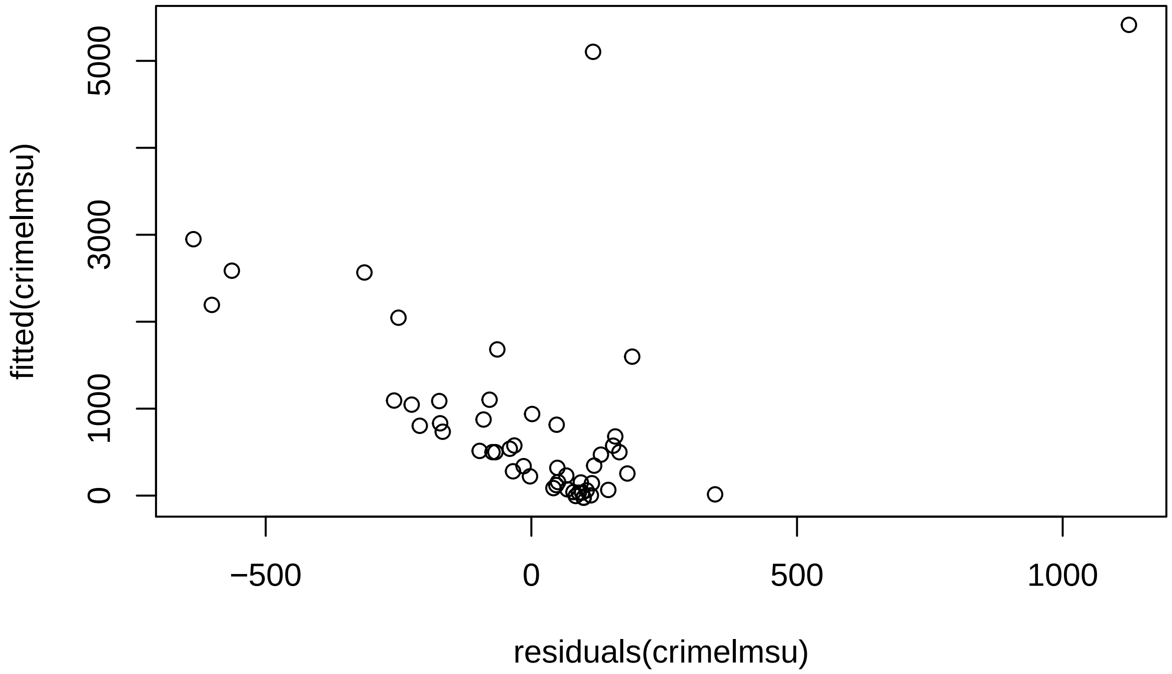
In result, we can't find any kind of linear structure, which would mean that we would need to include them in the model beacuse of the relation.

Next, we will scatter against the response variable.

```
plot(residuals(crimelmsu),expend)
```



```
plot(residuals(crimelmsu),fitted(crimelmsu))
```



The spread looks good.

And finally, we will perform a Saphiro test to check the normality assumption.

```
shapiro.test(residuals(crimelmsu))
```

As we can see above, the p-value for the test is really low, 2.11e-06. This means that the assumption of normality is, if only, doubtful.