EDDA - Assignment 2 - Group 77

Dante de Lang, Ignas Krikštaponis and Kamiel Gülpen

Exercise 1

Moldy bread If left alone bread will become moldy, rot or decay otherwise. To investigate the influence of temperature and humidity on this process, the time to decay was measured for 18 slices of white bread, which were placed in 3 different environments and humidified or not. The data are given in the file bread.txt, with the first column time to decay in hours, the second column the environment (cold, warm or intermediate temperature) and the third column the humidity.

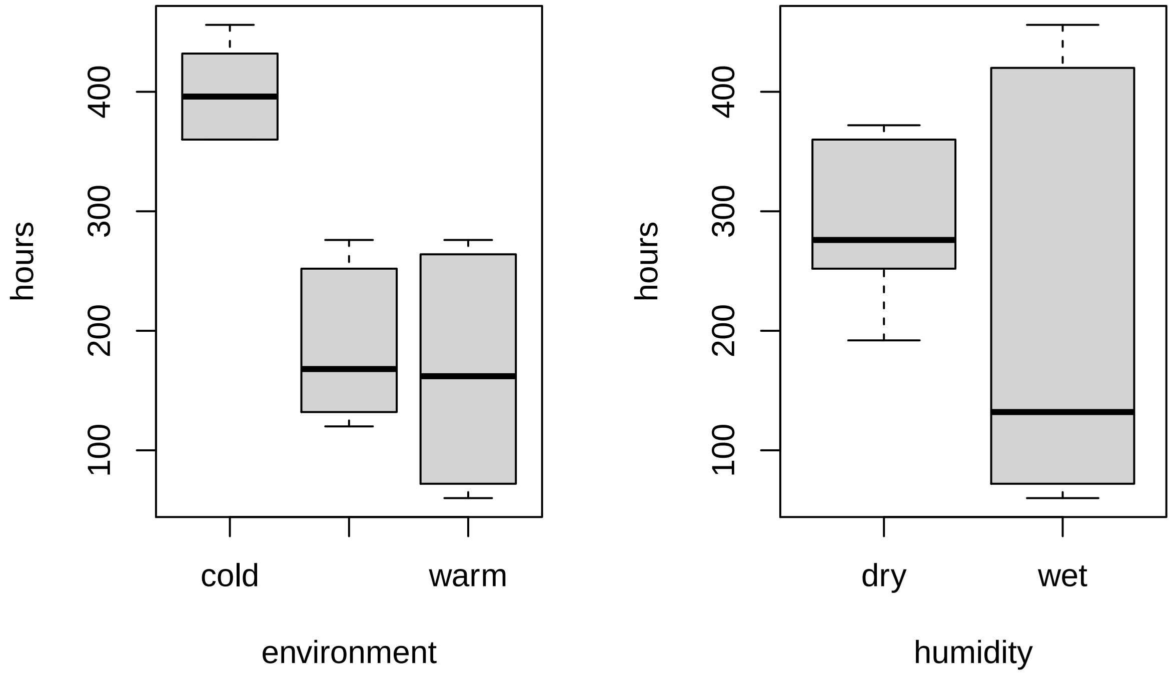
a)The 18 slices came from a single loaf, but were randomized to the 6 combinations of conditions. Present an R-code for this randomization process.

```
data <- read.table(file="data/bread.txt",header=TRUE)
I=3; J=2; N=3
env = rep(c("cold","intermediate","warm"), each=N*J)
hum = rep(c("dry", "wet"),each = N*I)
cbind(env, hum, sample(1:(N*I*J)))
```

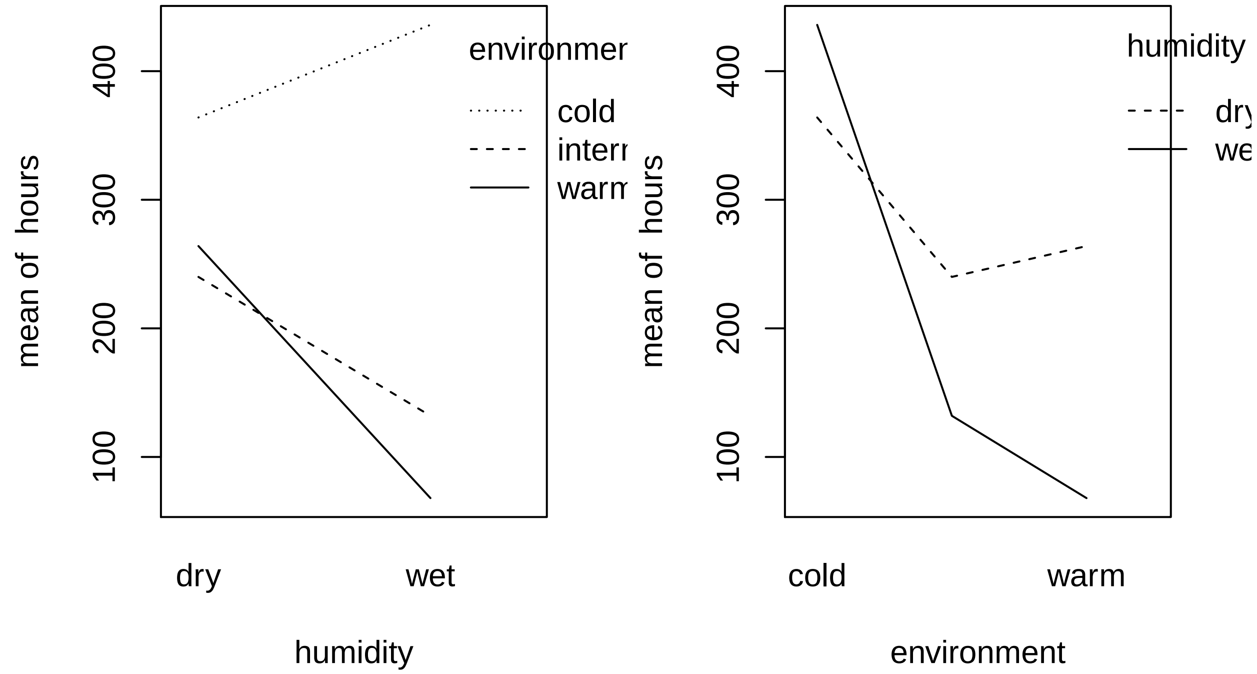
```
##       env            hum
##  [1,] "cold"         "dry" "1"
##  [2,] "cold"         "dry" "13"
##  [3,] "cold"         "dry" "10"
##  [4,] "cold"         "dry" "6"
##  [5,] "cold"         "dry" "4"
##  [6,] "cold"         "dry" "16"
##  [7,] "intermediate" "dry" "14"
##  [8,] "intermediate" "dry" "15"
##  [9,] "intermediate" "dry" "7"
## [10,] "intermediate" "wet" "3"
## [11,] "intermediate" "wet" "12"
## [12,] "intermediate" "wet" "9"
## [13,] "warm"         "wet" "17"
## [14,] "warm"         "wet" "2"
## [15,] "warm"         "wet" "18"
## [16,] "warm"         "wet" "11"
## [17,] "warm"         "wet" "8"
## [18,] "warm"         "wet" "5"
```

b)Make two boxplots of hours versus the two factors and two interaction plots (keeping the two factors fixed in turn).

```
par(mfrow=c(1,2))
attach(data)
boxplot(hours~environment)
boxplot(hours~humidity)
```



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



c)Perform an analysis of variance to test for effect of the factors temperature,humidity, and the interaction. Describe the interaction effect in words.

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

```
## Analysis of Variance Table
##
## Response: hours
##                      Df Sum Sq Mean Sq F value  Pr(>F)
## humidity              1  26912   26912    62.3 4.3e-06 ***
## environment           2 201904  100952   233.7 2.5e-10 ***
## humidity:environment  2  55984   27992    64.8 3.7e-07 ***
## Residuals            12   5184     432
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(dataaov)
```

```
##
## Call:
## lm(formula = hours ~ humidity * environment)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
##    -48     -7      0     11     36
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             364         12   30.33  1.0e-12 ***
## humiditywet                              72         17    4.24   0.0011 **
## environmentintermediate                -124         17   -7.31  9.4e-06 ***
## environmentwarm                        -100         17   -5.89  7.3e-05 ***
## humiditywet:environmentintermediate    -180         24   -7.50  7.2e-06 ***
## humiditywet:environmentwarm            -268         24  -11.17  1.1e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.8 on 12 degrees of freedom
## Multiple R-squared:  0.982,  Adjusted R-squared:  0.975
## F-statistic:  132 on 5 and 12 DF,  p-value: 4.68e-10
```

When looking at the two-way anova model we see that it consists of the following terms: Y_{ijk} = μ_{ij} + e_{ijk} = $\mu + alpha_i$ + β_j + γ_{ij} + e_{eijk} We decompose the formula it this way such that μ is the overall mean, α_i and β_j are the main effect of level i and j of the first factor and second factor respectively and γ_{ij} the interaction effect.

In order to test the effect of the temperature,humidity, and the interaction we set up 3 hypotheses which are: H_{AB}: $\gamma_{ij} = 0$ for every (i, j) (no interactions between factor A and B)

H_A: $\alpha_i = 0$ for every i (no main effect of factor A)

H_B:$\beta_j = 0$ for every j (no main effect of factor B)

We use the test statistics F_{AB} for H_{AB}, F_A for H_A and F_B for H_B where F is the F-distribution.

To see if the Hypotheses can be rejected we want to look at the probability that P(F>f_{AB}), P(F>f_A) and P(F>f_B), the bigger the F value the lower the probability that the Hypothesis lays under a F-distribution and therefore the Hypothesis can be rejected.

We see that the humidity has a p-value of 4.3e-06, environment a p-value of 2.5e-10 and the interaction between the two (humidity:environment) shows a p-value of 3.7e-07. This means that humidity, environment and the interaction effect between humidity and environment have a significant influence on the hours, which means we can reject H_A, H_B and H_{AB}.

The interaction effect looks at the difference of differences, for example: it looks at the difference in hours for environment = cold and environment = warm for humidity = wet. Then it looks the difference between environment = cold and environment = warm for humidity = dry. It then looks at the difference between those differences and when this difference is high it shows that there is indeed interaction.

d) Which of the two factors has the greatest (numerical) influence on the decay? Is this a good question?

```
# Without interaction
data$humidity=as.factor(data$humidity)
data$environment=as.factor(data$environment)
dataaov=lm(hours~humidity+environment,data=data)
anova(dataaov)
```

```
## Analysis of Variance Table
##
```

```
## Response: hours
##             Df Sum Sq Mean Sq F value  Pr(>F)
## humidity     1  26912   26912    6.16   0.026 *
## environment  2 201904  100952   23.11 3.7e-05 ***
## Residuals   14  61168    4369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

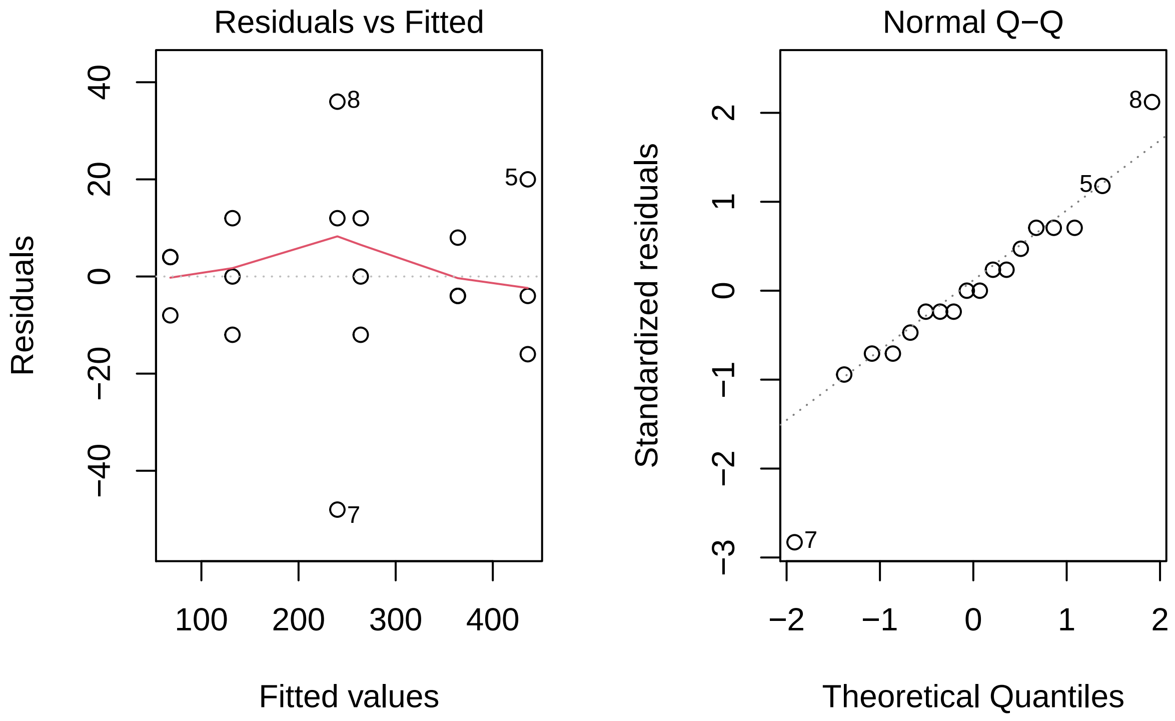
When we want to know which factor has the greatest influence we want to use the additive model as used above. This shows a p-value of 0.026 for humidity and a p-value of 3.7e-05 for environemnt. This means that the environment has the greatest influence.

e) Check the model assumptions by using relevant diagnostic tools. Are there any outliers?

```
par(mfrow=c(1,2))
dataaov2=lm(hours~humidity*environment,data=data);
shapiro.test(residuals(dataaov2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(dataaov2)
## W = 0.9, p-value = 0.2
```

```
plot(dataaov2, 1)
plot(dataaov2, 2)
```



The qqplot shows a somewhat linear line which means that based on the qqplot we can state that the data is normally distributed. Furthermore we used a Shapiro-Wilks test to see if the test can back this assumption. The Shapiro-Wils test showed a p-value of 0.2 which means that the data is normally distributed. There is also looked at the spread of the residuals, which showed that there are three outliers which are number 5, 7 and 8 which can be observed in both plot.

Exercise 2

A researcher is interested in the time it takes a student to find a certain product on the internet using a search engine. There are three different types of interfaces with the search engine and especially the effect of these interfaces is of importance. There are five different types of students, indicating their level of computer skill (the lower the value of this indicator, the better the computer skill of the corresponding student). Fifteen students are selected; three from each group with a certain level of computer skill. The data is given in the file search.txt. Assume that the experiment was run according to a randomized block design which you make in a). (Beware that the levels of the factors are coded by numbers.)

a) Number the selected students 1 to 15 and show how (by using R) the students could be randomized to the interfaces in a randomized block design.

```
interface <- factor(rep(c(1,2,3),each = 5))
skill <- factor(rep(c(1,2,3,4,5),times = 3))
students <- c(1:15)
block <- data.frame(students,skill,interface); block
```

```
##    students skill interface
## 1         1     1         1
## 2         2     2         1
## 3         3     3         1
## 4         4     4         1
## 5         5     5         1
## 6         6     1         2
## 7         7     2         2
## 8         8     3         2
## 9         9     4         2
## 10       10     5         2
## 11       11     1         3
## 12       12     2         3
## 13       13     3         3
## 14       14     4         3
## 15       15     5         3
```

```
# REMARK: Do i need to include skill, since only interfaces is mentioned. And perhaps use a random samp
```

b) Test the null hypothesis that the search time is the same for all interfaces. What type of interface does require the longest search time? For which combination of skill level and type of interface is the search time the shortest? Estimate the time it takes a typical user of skill level 3 to find the product on the website if the website uses interface 3.

```
data_search <- read.table(file="data/search.txt",header=TRUE)
data_search$skill <- as.factor(data_search$skill)
data_search$interface <- as.factor(data_search$interface)

aovsearch = lm(data_search$time~data_search$interface+data_search$skill, data= data_search)
```

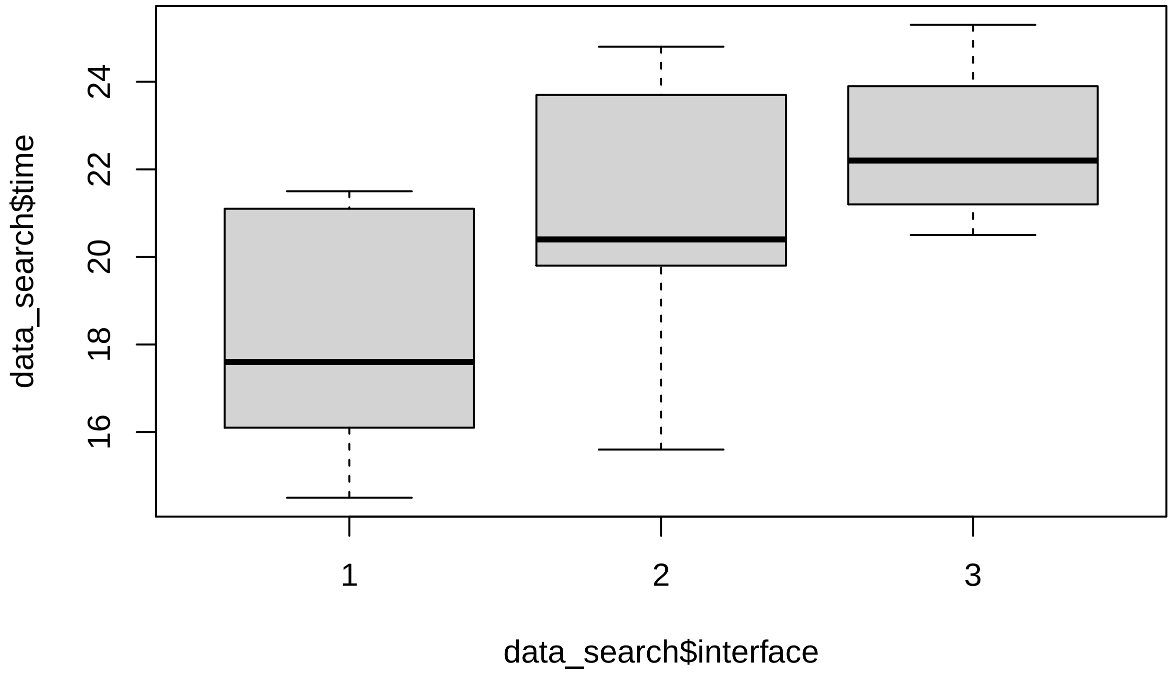
```
anova(aovsearch) # The p-value for the interfaces is not significant >0.05 and therefore search time of
```

```
## Analysis of Variance Table
##
## Response: data_search$time
##                         Df Sum Sq Mean Sq F value Pr(>F)
## data_search$interface  2   50.5   25.23    7.82  0.013 *
## data_search$skill      4   80.1   20.01    6.21  0.014 *
## Residuals              8   25.8    3.23
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aovsearch)
```

```
##
## Call:
## lm(formula = data_search$time ~ data_search$interface + data_search$skill,
##     data = data_search)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.573 -0.697  0.387  1.057  1.787
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)               15.01       1.23   12.24  1.8e-06 ***
## data_search$interface2     2.70       1.14    2.38   0.0447 *
## data_search$interface3     4.46       1.14    3.93   0.0044 **
## data_search$skill2         1.30       1.47    0.89   0.4012
## data_search$skill3         3.03       1.47    2.07   0.0724 .
## data_search$skill4         5.30       1.47    3.61   0.0068 **
## data_search$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
```

```
boxplot(data_search$time~data_search$interface) # Interface 3 has the longest search time
```

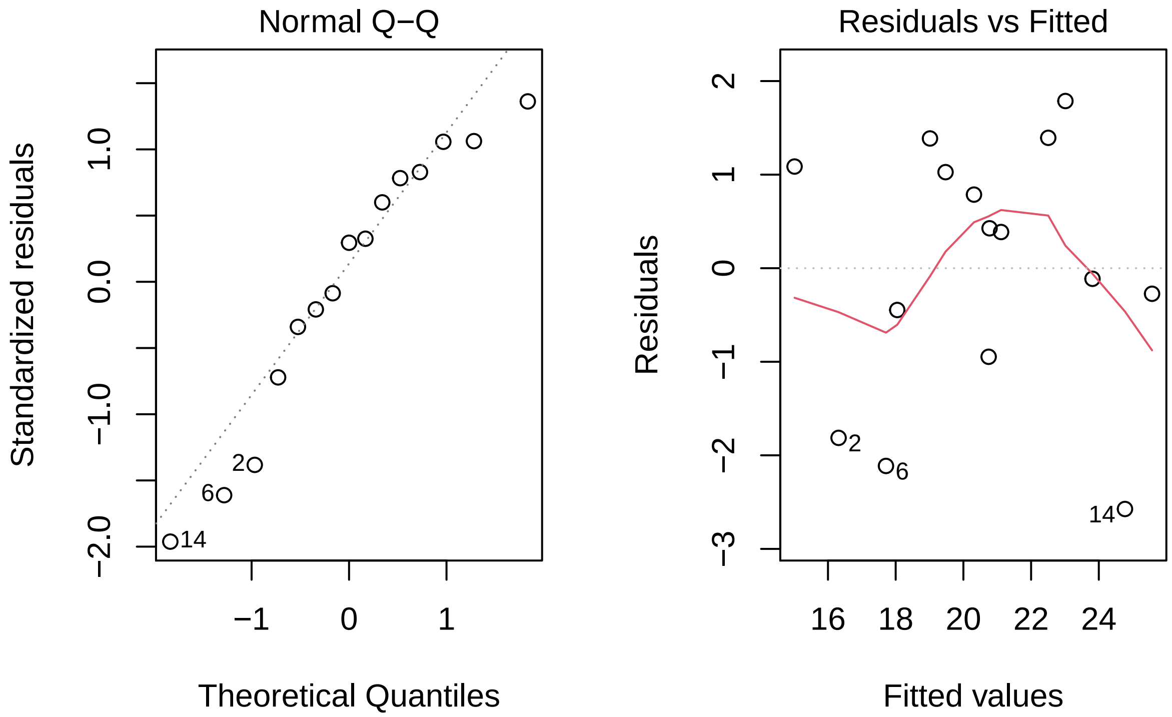


```
 # Skill 2 and interface 1 is the fastest

# Estimate interface 3 = 4.46, skill 3 = 3.03, so 3-3 gives:
(4.46+3.03)/2 # 3.75 seconds ???? Still vague how this works
```

```
## [1] 3.75
```

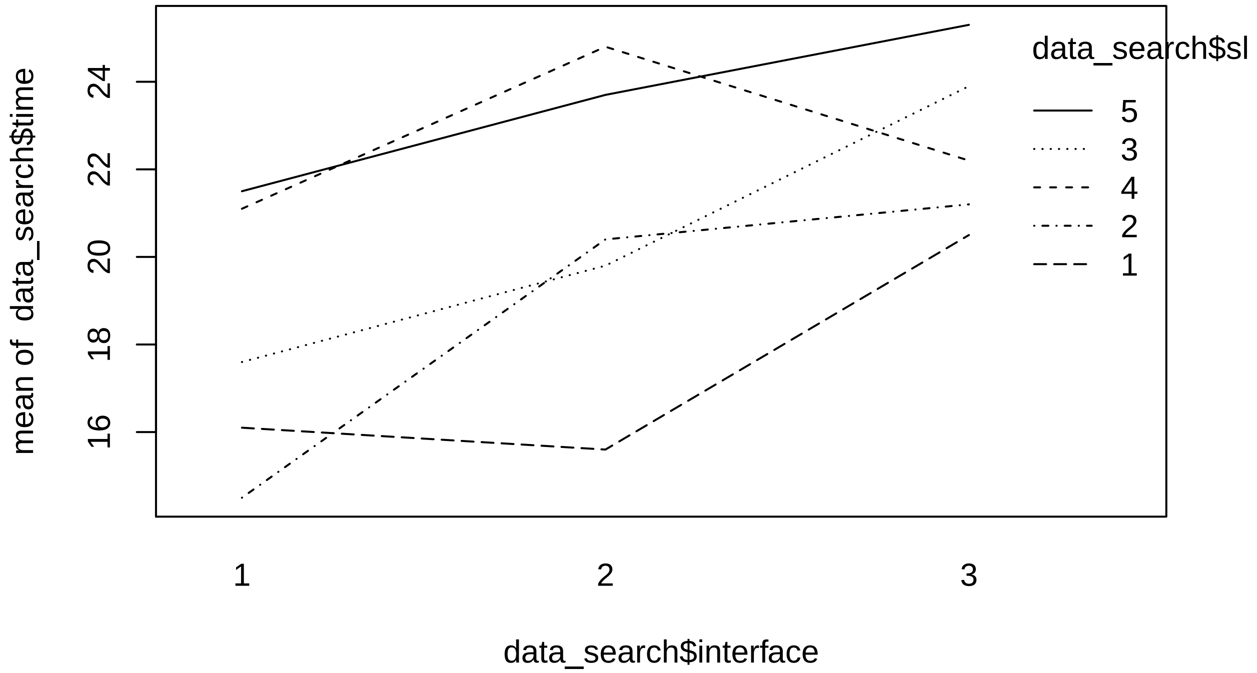
c) Check the model assumptions by using relevant diagnostic tools.

```
par(mfrow=c(1,2))
# qqnorm(residuals(aovsearch))
# plot(fitted(aovsearch),residuals(aovsearch))
plot(aovsearch,2)
plot(aovsearch,1)
```



d) Perform the Friedman test tot test whether there is an effect of interface.

```
interaction.plot(data_search$interface,data_search$skill,data_search$time) # Parallel lines indicate no
```



```
friedman.test(data_search$time,data_search$interface,data_search$skill) # P-value is significant thus H
```

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

e)Test the null hypothesis that the search time is the same for all interfaces by a one-way ANOVA test, ignoring the variable skill. Is it right/wrong or useful/not useful to perform this test on this dataset?

```
aovsearch = lm(data_search$time~data_search$interface)
anova(aovsearch)
```

```
## Analysis of Variance Table
##
## Response: data_search$time
##                         Df Sum Sq Mean Sq F value Pr(>F)
## data_search$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
```

```
# is it not useless also to ignore skill since the time is clearly also depended on this variable, you

# While only looking at interface with I/degrees = 2, you can also just perform a t.test, no????

t.test(data_search$time,as.numeric(data_search$interface))
```

```
##
##  Welch Two Sample t-test
##
## data:  data_search$time and as.numeric(data_search$interface)
## t = 21, df = 16, p-value = 7e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  16.7 20.4
## sample estimates:
## mean of x mean of y
##      20.5       2.0
```

Excercise 3

In a study on the effect of feedingstuffs on lactation a sample of nine cows were fed with two types of food, and their milk production was measured. All cows were fed both types of food, during two periods, with a neutral period in-between to try and wash out carry-over effects. The order of the types of food was randomized over the cows. The observed data can be found in the file cow.txt, where A and B refer to the types of feedingstuffs.

a) Test whether the type of feedingstuffs influences milk production using an ordinary "fixed effects" model, fitted with lm. Estimate the difference in milk production.

```
# read data
data <- read.table(file="data/cow.txt",header=TRUE)
data$treatment <- as.factor(data$treatment); data$order <- as.factor(data$order)
data$id <- as.factor(data$id); data$per <- as.factor(data$per)

# perform fixed effects model analysis
fixed_aov <- lm(milk ~ order + id + per + treatment, data = data)
summary(fixed_aov)
```

```
##
## Call:
## lm(formula = milk ~ order + id + per + treatment, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.260 -0.438  0.000  0.438  2.260
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   30.300      1.244   24.35  5.0e-08 ***
## orderBA      -11.200      1.574   -7.12  0.00019 ***
## id2           23.000      1.574   14.61  1.7e-06 ***
## id3           11.150      1.574    7.08  0.00020 ***
## id4           -1.350      1.574   -0.86  0.41948
## id5            4.150      1.574    2.64  0.03360 *
## id6           34.650      1.574   22.01  1.0e-07 ***
## id7           24.750      1.574   15.72  1.0e-06 ***
## id8           16.100      1.574   10.23  1.8e-05 ***
## id9               NA         NA      NA       NA
## per2          -2.390      0.747   -3.20  0.01505 *
## treatmentB    -0.510      0.747   -0.68  0.51654
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.57 on 7 degrees of freedom
## Multiple R-squared:  0.993,  Adjusted R-squared:  0.983
## F-statistic:  101 on 10 and 7 DF,  p-value: 1.35e-06
```

b)

```
attach(data)
mixed_avo <- lmer(milk ~ treatment + order + per + (1|id),REML=FALSE)
mixed_avo_1 <- lmer(milk ~ order + per + (1|id),REML=FALSE)
anova(mixed_avo_1, mixed_avo)
```

```
## Data: NULL
## Models:
## mixed_avo_1: milk ~ order + per + (1 | id)
## mixed_avo: milk ~ treatment + order + per + (1 | id)
##             npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)
## mixed_avo_1    5 118 122  -53.9      108
## mixed_avo      6 119 125  -53.7      107  0.58  1       0.45
```

c)

```
attach(data)
```

```
## The following objects are masked from data (pos = 3):
##
##     id, milk, order, per, treatment
```

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

```
##
##  Paired t-test
##
## data:  milk[treatment == "A"] and milk[treatment == "B"]
## t = 0.2, df = 8, p-value = 0.8
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.27  2.76
## sample estimates:
## mean of the differences
##                   0.244
```

Exercise 4

Stochastic models for word counts are used in quantitative studies on literary styles. Statistical analysis of the counts can, for example, be used to solve controversies about true author ships. Another example is the analysis of word frequencies in relation to Jane Austen's novel Sanditon. At the time Austen died, this novel was only partly completed. Austen, however, had made a summary for the remaining part. An admirer of Austen's work finished the novel, imitating Austen's style as much as possible. The file austen.txt contains counts of different words in some of Austen's novels: chapters 1 and 3 of Sense and Sensibility(stored in the Sense column), chapters 1, 2 and 3 of Emma(column Emma), chapters 1 and 6 of Sanditon(both written by Austen herself, column Sand1) and chapters 12 and 24 of Sanditon(both written by the admirer,Sand2)

a) Discuss whether a contingency table test for independence or for homogeneity is most appropriate here.

The contingency table test for homogeneity is appropriate because we want to know if the fan writer imitates Austen in a good way. This means that we want to test whether or not the different columns of data in the table come from the same population (writer) or not, which would be the case it the fan imitated Austen correctly. The H0 of the contingency table test for homogeneity states that the distribution of the words is the same for the stories.

b) Using the given data set, investigate whether Austen herself was consistent in her different novels. Where are the main inconsistencies?

```
data=read.table(file="data/austen.txt",header=TRUE)
austen = data[,1:3]
z = chisq.test(austen)
z
```

```
##
##  Pearson's Chi-squared test
##
## data:  austen
## X-squared = 12, df = 10, p-value = 0.3
```

```
residuals(z)
```

```
##           Sense   Emma  Sand1
## a       -1.0300 -0.129  1.594
## an       0.4473 -0.159 -0.375
## this     0.0513  0.294 -0.504
## that     0.7482  0.287 -1.442
## with    -0.0475  0.521 -0.704
## without  1.0654 -1.588  0.893
```

She is not inconsistent as the p-value is above 0.05. This means that we cannot reject the H0. She does however have some main inconsistency, which where the words "a", "that" and "without". As can be seen in the residual tanle above.

```
z = chisq.test(data)
z
```

```
##
##  Pearson's Chi-squared test
##
## data:  data
## X-squared = 46, df = 15, p-value = 6e-05
```

```
residuals(z)
```

```
##            Sense      Emma  Sand1   Sand2
## a       -1.015 -0.112093  1.606 -0.0589
## an      -0.591 -1.219955 -1.067  3.7282
## this     0.139  0.390490 -0.444 -0.3267
## that     1.594  1.179849 -0.910 -3.0493
## with    -0.512  0.000192 -1.025  1.7482
## without  1.392 -1.341196  1.137 -1.0696
```

The fan is inconsistent as the p-value of the test is below 0.05. Therefore we have to reject the H0 and accept that the distribution of the words in the stories are not the same. Because Austen herself did not have this inconsistency we can say that the inconsistency is caused by the fan writer. The main inconsistencies were for the words "that" and "an". As can be seen in the residual tanle above.

Exercise 5

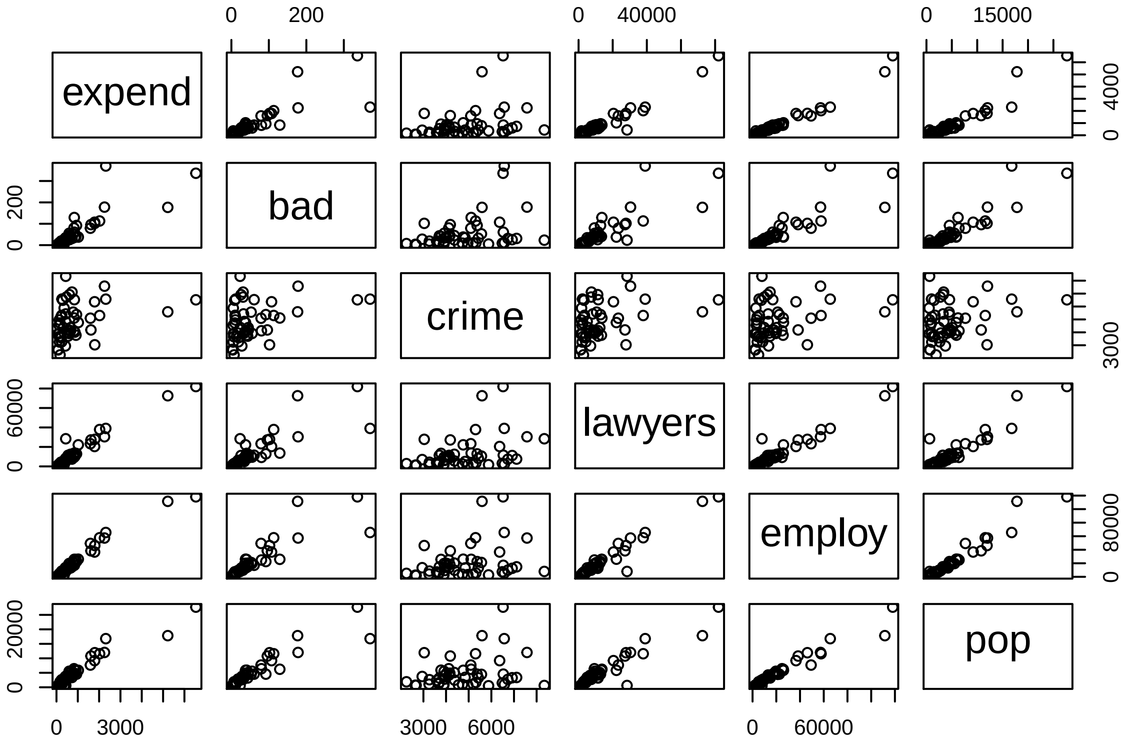
The data in expenses crime.txt were obtained to determine factors related to state expenditures on criminal activities (courts, police, etc.) The variables are:state(indicating the state in the USA),expend(state expenditures on criminal activities in$1000),bad(crime rate per 100000),crime(number of persons under criminal supervision),lawyers(number of lawyers in the state),employ(number of persons employed in the state) and pop(population of the state in 1000). In the regression analysis, take expend as response variable and bad,crime,lawyers,employ and pop as explanatory variables.

a) Make some graphical summaries of the data. Investigate the problem of potential and influence points,and the problem of collinearity.

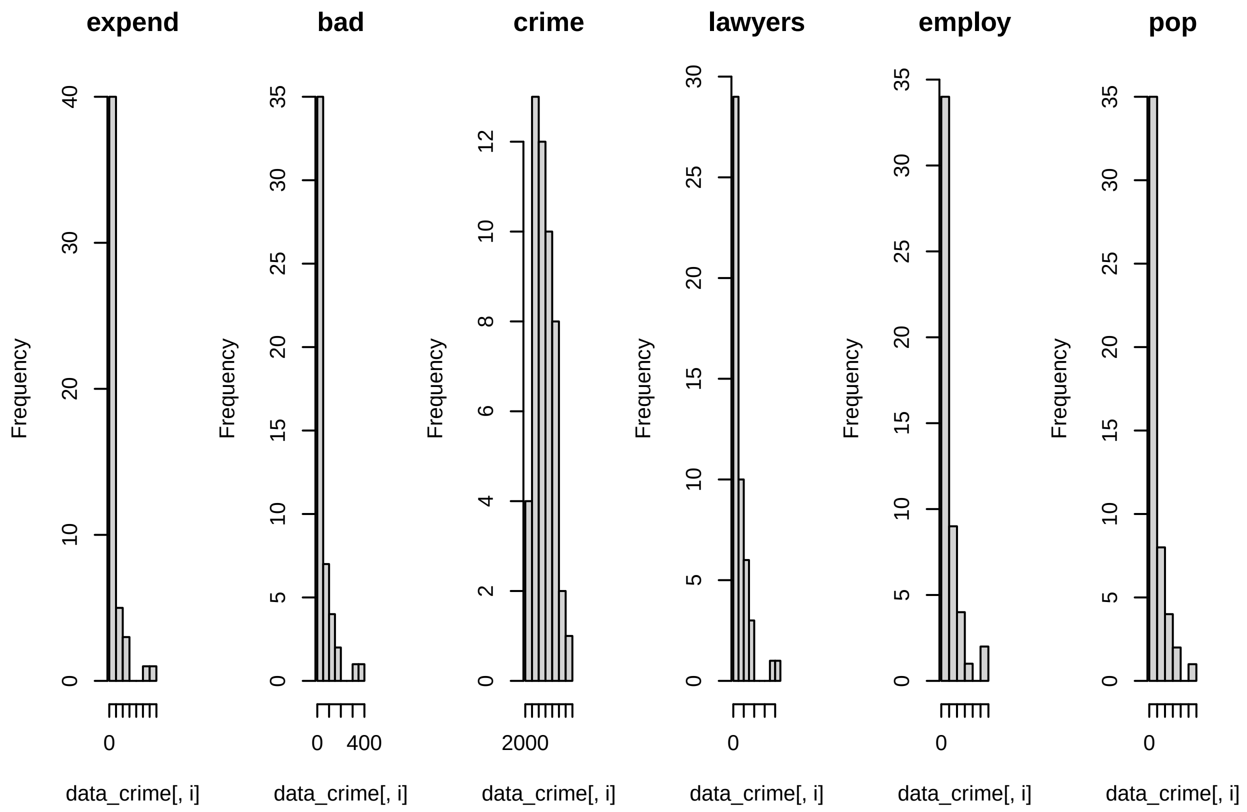
```
data_crime = read.table(file="data/expensescrime.txt",header=TRUE)
data_crime
```

```
##    state expend   bad crime lawyers employ   pop
## 1     AK    360   5.1  5877    1749   2796   525
## 2     AL    498  34.4  3942    6679  13999  4083
## 3     AR    219  19.2  3585    3741   7227  2388
## 4     AZ    728  31.3  7116    7535  14755  3386
## 5     CA   6539 336.2  6518   82001 118149 27663
## 6     CO    602  25.7  6919   11174  12556  3296
## 7     CT    544  43.5  3705   11397  14798  3211
## 8     DC    435  23.3  8339   28399   7925   622
## 9     DE    130  10.6  4961    1597   3230   644
## 10    FL   2252 177.9  7574   30444  57310 12023
## 11    GA    835 129.2  5110   13652  25848  6222
## 12    HI    210  10.8  5201    2787   3886  1083
## 13    IA    368  17.7  3943    6182   9309  2834
## 14    ID    120   5.8  3908    2031   3363   998
## 15    IL   2023 113.0  5303   37873  57748 11582
## 16    IN    593  55.3  3914    9499  19647  5531
## 17    KS    324  23.8  4375    5555   9726  2476
## 18    KY    417  27.9  2947    7017  13480  3727
## 19    LA    785  52.7  5564   10569  21184  4461
## 20    MA   1024  37.8  4758   22154  26048  5855
## 21    MD    940  92.0  5373   12866  22541  4535
## 22    ME    128   6.3  3672    2528   4340  1187
## 23    MI   1788 107.2  6366   20445  36632  9200
## 24    MN    665  38.6  4134   11343  13159  4246
## 25    MO    660  44.9  4366   12439  20260  5103
## 26    MS    245  18.9  3266    4270   8463  2625
## 27    MT    123   4.9  4549    2006   3211   809
## 28    NC    821  80.2  4121    9265  24843  6413
## 29    ND     75   2.4  2679    1290   1997   672
## 30    NE    206  13.7  3695    4289   5820  1594
## 31    NH    140   4.8  3252    2139   4034  1057
## 32    NJ   1592  79.2  5094   23301  49346  7672
## 33    NM    296   8.9  6486    3164   7413  1500
## 34    NV    256  11.4  6575    2276   5528  1007
## 35    NY   5220 176.7  5589   72575 111518 17825
## 36    OH   1617  96.0  4187   27191  38404 10784
## 37    OK    432  32.4  5425    8302  13167  3272
## 38    OR    463  31.2  6730    7385   9858  2724
## 39    PA   1796 101.9  3037   27798  46200 11936
## 40    RI    164   9.2  4723    2527   3774   986
## 41    SC    427  34.5  4841    5021  13177  3425
## 42    SD     79   3.9  2641    1230   2396   709
## 43    TN    568  45.2  4167    8782  18190  4855
## 44    TX   2313 370.1  6569   39028  65488 16789
## 45    UT    244  10.0  5317    3446   5715  1680
## 46    VA    914  40.5  3779   13390  25720  5904
## 47    VT     74   6.2  3888    1372   1969   548
## 48    WA    838  60.7  6529   11507  17020  4538
## 49    WI    863  36.6  4017   10316  19911  4807
## 50    WV    168   7.2  2253    2835   5079  1897
```

```
plot(data_crime[,c(2,3,4, 5, 6, 7)])
```



```
par(mfrow=c(1,6))
for (i in c(2,3,4, 5, 6, 7)) hist(data_crime[,i],main=names(data_crime)[i])
```



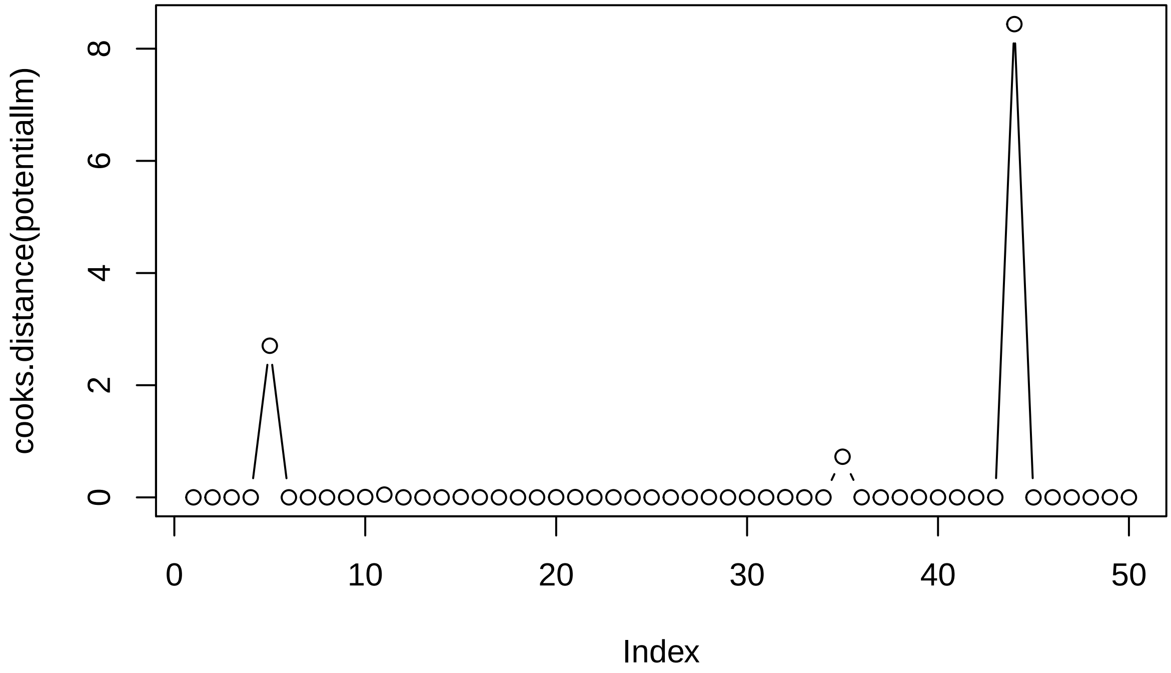
```
regression_data = data_crime[2:7]

par(mfrow=c(1,1))
potentiallm = lm(expend~bad, data = regression_data)
potentiallm
```

```
##
## Call:
## lm(formula = expend ~ bad, data = regression_data)
##
## Coefficients:
## (Intercept)          bad
##       128.3         13.3
```

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

```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 0.00 0.00 0.00 0.00 2.70 0.00 0.00 0.00 0.00 0.01 0.05 0.00 0.00 0.00 0.01 0.00
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
## 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
## 0.00 0.00 0.73 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 8.44 0.00 0.00 0.00 0.00
##   49   50
## 0.00 0.00
```

```
plot(cooks.distance(potentiallm), type="b")
```



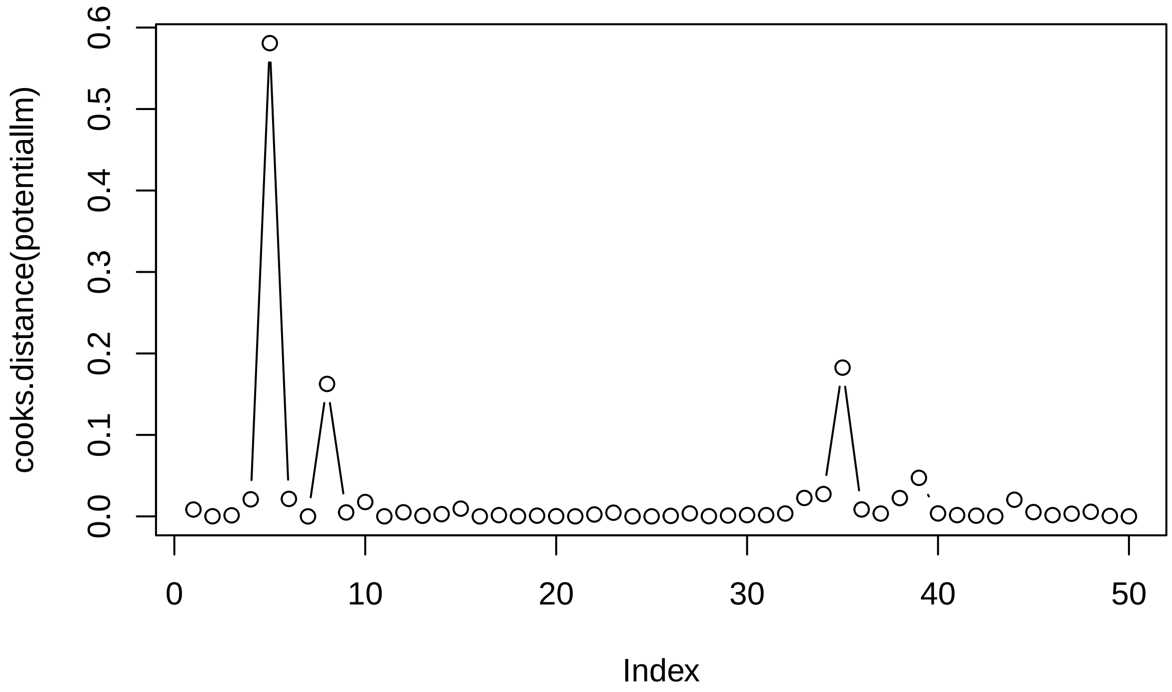
```
potentiallm = lm(expend~crime, data = regression_data)
potentiallm
```

```
##
## Call:
## lm(formula = expend ~ crime, data = regression_data)
##
## Coefficients:
## (Intercept)        crime
##    -500.284        0.283
```

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

```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 0.01 0.00 0.00 0.02 0.58 0.02 0.00 0.16 0.00 0.02 0.00 0.01 0.00 0.00 0.01 0.00
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
## 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
## 0.02 0.03 0.18 0.01 0.00 0.02 0.05 0.00 0.00 0.00 0.00 0.02 0.01 0.00 0.00 0.01
##   49   50
## 0.00 0.00
```

```
plot(cooks.distance(potentiallm), type="b")
```



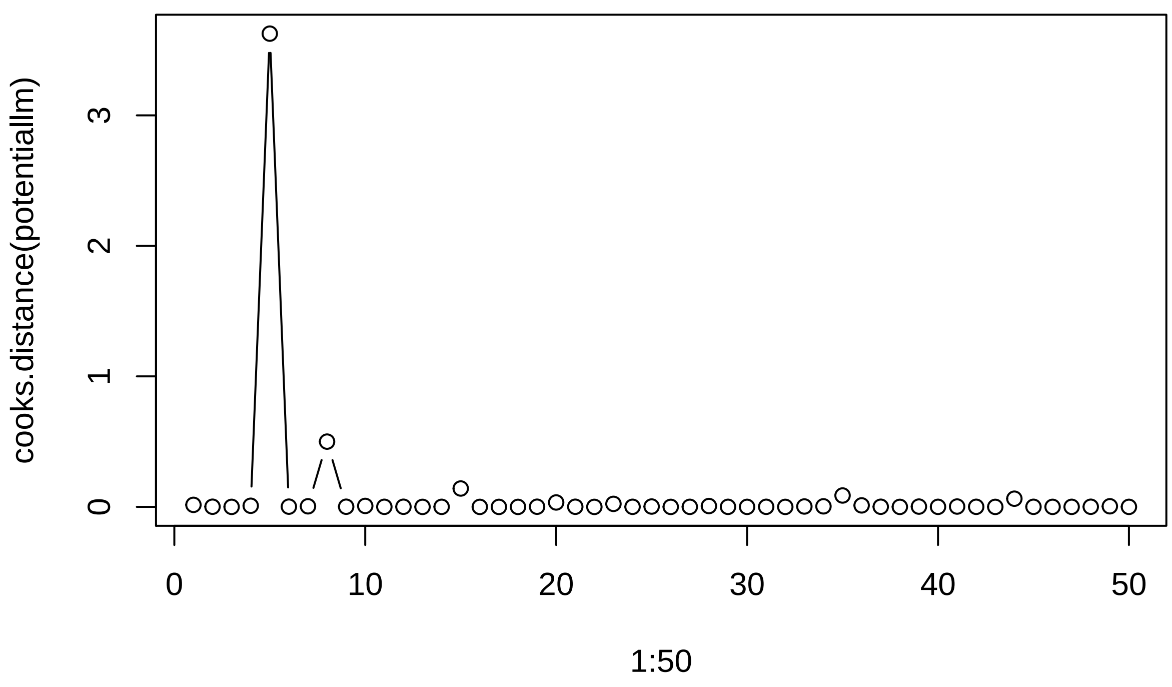
```
potentiallm = lm(expend~lawyers, data = regression_data)
potentiallm
```

```
##
## Call:
## lm(formula = expend ~ lawyers, data = regression_data)
##
## Coefficients:
## (Intercept)      lawyers
##    -62.6806       0.0705
```

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

```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 0.02 0.00 0.00 0.01 3.63 0.00 0.00 0.50 0.00 0.01 0.00 0.00 0.00 0.00 0.14 0.00
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
## 0.00 0.00 0.00 0.03 0.00 0.00 0.02 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.00
##   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
## 0.00 0.00 0.09 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.06 0.00 0.00 0.00 0.00
##   49   50
## 0.00 0.00
```

```
plot(1:50,cooks.distance(potentiallm), type="b")
```



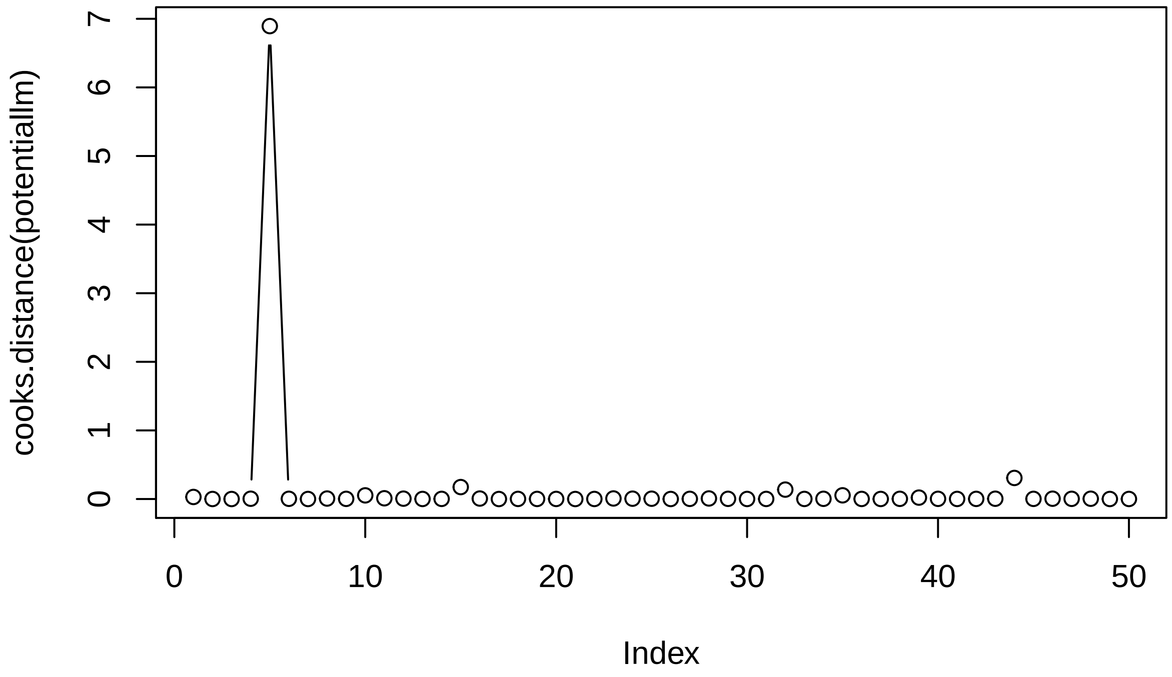
```
potentiallm = lm(expend~employ, data = regression_data)
potentiallm
```

```
##
## Call:
## lm(formula = expend ~ employ, data = regression_data)
##
## Coefficients:
## (Intercept)       employ
##   -120.3669       0.0469
```

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

```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 0.03 0.00 0.00 0.00 6.89 0.00 0.00 0.01 0.00 0.05 0.01 0.01 0.00 0.00 0.17 0.01
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
## 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.14
##   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
## 0.00 0.00 0.05 0.00 0.00 0.00 0.02 0.00 0.00 0.00 0.00 0.31 0.00 0.00 0.00 0.00
##   49   50
## 0.00 0.00
```

```
plot(cooks.distance(potentiallm), type="b")
```



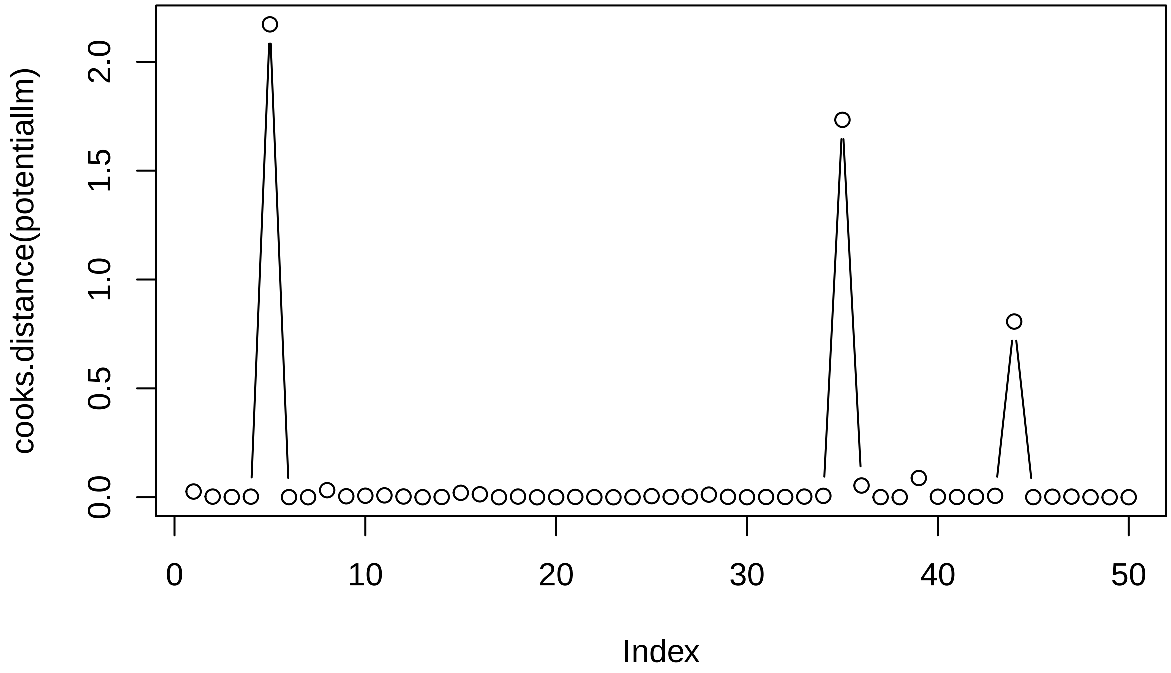
```
potentiallm = lm(expend~pop, data = regression_data)
potentiallm
```

```
##
## Call:
## lm(formula = expend ~ pop, data = regression_data)
##
## Coefficients:
## (Intercept)          pop
##    -195.844        0.218
```

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

```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 0.03 0.00 0.00 0.00 2.17 0.00 0.00 0.03 0.00 0.01 0.01 0.00 0.00 0.00 0.02 0.01
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
## 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.01 0.00 0.00 0.00 0.00
##   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
## 0.00 0.01 1.73 0.05 0.00 0.00 0.09 0.00 0.00 0.00 0.01 0.81 0.00 0.00 0.00 0.00
##   49   50
## 0.00 0.00
```

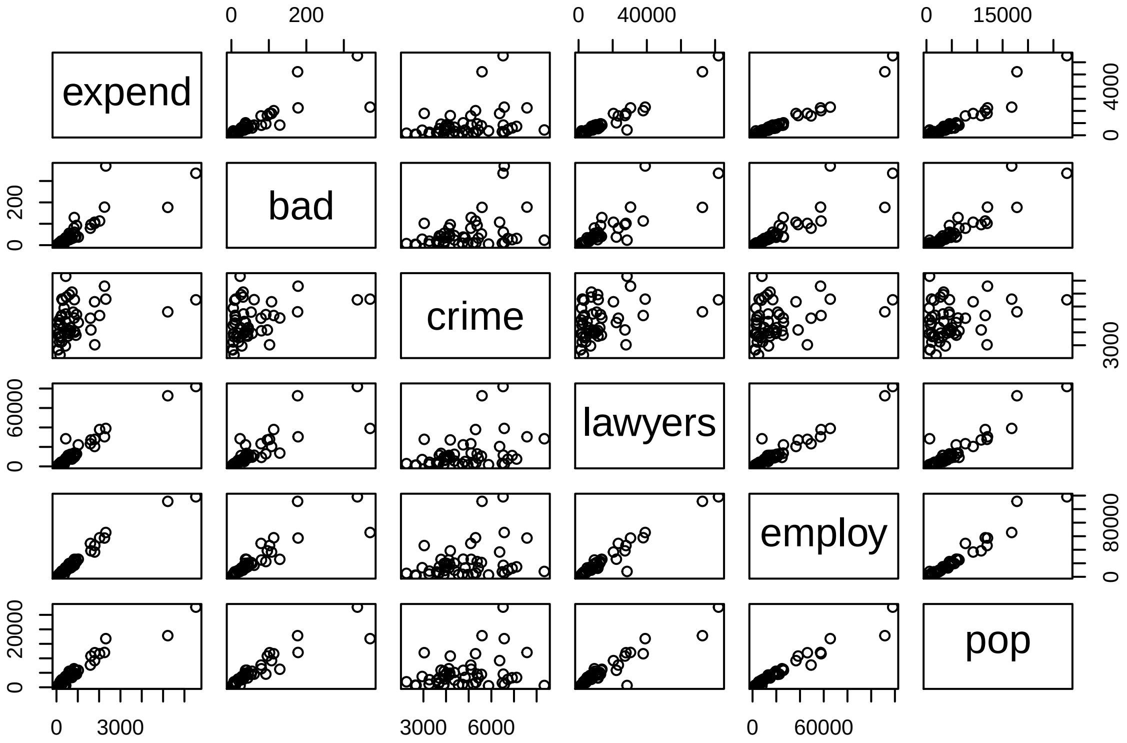
```
plot(cooks.distance(potentiallm), type="b")
```



```
# Collinearity
round(cor(regression_data),2)
```

```
##          expend  bad crime lawyers employ  pop
## expend     1.00 0.83  0.33    0.97   0.98 0.95
## bad        0.83 1.00  0.37    0.83   0.87 0.92
## crime      0.33 0.37  1.00    0.37   0.30 0.27
## lawyers    0.97 0.83  0.37    1.00   0.97 0.93
## employ     0.98 0.87  0.30    0.97   1.00 0.97
## pop        0.95 0.92  0.27    0.93   0.97 1.00
```

```
pairs(regression_data)
```



```
# We see that employee and and lawyers are strongly correlated(0.97)
# We see that employee and crime rate per 100000 are strongly correlated(0.87)
# We see that lawyers and  crime rate per 100000 are strongly correlated(0.83)
# We see a correlation betwen pop and bad and pop and lawyers and pop and employ

regressionlm=lm(expend~bad+crime+lawyers+employ, data=regression_data)
car::vif(regressionlm)
```

```
## Registered S3 methods overwritten by 'car':
##   method                          from
##   influence.merMod                lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod         lme4
##   dfbetas.influence.merMod        lme4

##     bad   crime lawyers  employ
##    4.42    1.30   16.58   20.87
```

```
# We see a value above 5 for lawyers and employees which means we need to take one out

regressionlm=lm(expend~bad+crime+lawyers, data=regression_data)
car::vif(regressionlm)
```

```
##     bad   crime lawyers
##    3.26    1.17    3.27
```

```
# Now it looks good
```

b) Fit a linear regression model to the data. Use both the step-up and the step-down method to find thebest model. If step-up and step-down yield two different models, choose one and motivate your choice.

```
# Step-up method

summary(lm(expend~bad, data=regression_data)) #0.694
```

```
##
## Call:
## lm(formula = expend ~ bad, data = regression_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2742.7  -133.1   -75.6   110.9  2739.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   128.34     117.77    1.09     0.28
## bad            13.31       1.28   10.43  6.2e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 668 on 48 degrees of freedom
## Multiple R-squared:  0.694,  Adjusted R-squared:  0.688
## F-statistic:  109 on 1 and 48 DF,  p-value: 6.17e-14
```

```
summary(lm(expend~crime, data=regression_data)) #0.1
```

```
##
## Call:
## lm(formula = expend ~ crime, data = regression_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1423   -583   -181    138   5196
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -500.284    585.908   -0.85    0.397
## crime          0.283      0.117    2.42    0.019 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1140 on 48 degrees of freedom
## Multiple R-squared:  0.109,  Adjusted R-squared:  0.0901
## F-statistic: 5.85 on 1 and 48 DF,  p-value: 0.0194
```

```
summary(lm(expend~lawyers, data=regression_data)) #0.9369
```

```
##
## Call:
## lm(formula = expend ~ lawyers, data = regression_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1503.7   -28.9    36.3    94.5   822.9
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -62.68063   55.13018   -1.14     0.26
## lawyers       0.07047    0.00264   26.70   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 303 on 48 degrees of freedom
## Multiple R-squared:  0.937,  Adjusted R-squared:  0.936
## F-statistic:  713 on 1 and 48 DF,  p-value: <2e-16
```

```
summary(lm(expend~employ, data=regression_data))#0.954
```

```
##
## Call:
## lm(formula = expend ~ employ, data = regression_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -636.8  -85.0   50.1  106.1 1120.3
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.20e+02   4.82e+01    -2.5    0.016 *
## employ       4.69e-02   1.49e-03    31.5   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 260 on 48 degrees of freedom
## Multiple R-squared:  0.954,  Adjusted R-squared:  0.953
## F-statistic:  991 on 1 and 48 DF,  p-value: <2e-16
```

```
summary(lm(expend~pop, data=regression_data)) # 0.907
```

```
##
## Call:
## lm(formula = expend ~ pop, data = regression_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1148.3  -161.1    26.1   138.1  1533.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -195.8440    71.3698   -2.74   0.0085 **
## pop            0.2178     0.0101   21.66   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 368 on 48 degrees of freedom
## Multiple R-squared:  0.907,  Adjusted R-squared:  0.905
## F-statistic:  469 on 1 and 48 DF,  p-value: <2e-16
```

```
summary(lm(expend~employ+bad, data=regression_data))
```

```
##
## Call:
## lm(formula = expend ~ employ + bad, data = regression_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -655.3 -100.0   39.1  102.3 1149.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.20e+02   4.81e+01   -2.49    0.016 *
## employ       4.97e-02   3.01e-03   16.49   <2e-16 ***
## bad         -1.08e+00   1.00e+00   -1.08    0.286
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 259 on 47 degrees of freedom
## Multiple R-squared:  0.955,  Adjusted R-squared:  0.953
## F-statistic:  498 on 2 and 47 DF,  p-value: <2e-16
```

```
summary(lm(expend~employ+crime, data=regression_data))
```

```
##
## Call:
## lm(formula = expend ~ employ + crime, data = regression_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -666.9  -84.3   56.7  101.4 1119.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.56e+02   1.33e+02   -1.92    0.061 .
## employ       4.64e-02   1.56e-03   29.71   <2e-16 ***
## crime        3.03e-02   2.79e-02    1.09    0.282
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 259 on 47 degrees of freedom
## Multiple R-squared:  0.955,  Adjusted R-squared:  0.953
## F-statistic:  498 on 2 and 47 DF,  p-value: <2e-16
```

```
summary(lm(expend~employ+pop, data=regression_data))
```

```
##
## Call:
## lm(formula = expend ~ employ + pop, data = regression_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -689.4  -96.3   46.2  113.2 1065.1
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.31e+02   5.15e+01   -2.55    0.014 *
## employ       4.31e-02   6.21e-03    6.94    1e-08 ***
## pop          1.84e-02   2.96e-02    0.62    0.538
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 261 on 47 degrees of freedom
## Multiple R-squared:  0.954,  Adjusted R-squared:  0.952
## F-statistic:  489 on 2 and 47 DF,  p-value: <2e-16
```

```
summary(lm(expend~employ+lawyers, data=regression_data)) #0.9631 ==> only significant model
```

```
##
## Call:
## lm(formula = expend ~ employ + lawyers, data = regression_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -599.8  -93.4   38.4   94.8  931.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.15e+02   4.36e+01   -2.63   0.0115 *
## employ       2.98e-02   5.15e-03    5.77  5.9e-07 ***
## lawyers      2.69e-02   7.82e-03    3.44   0.0012 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 234 on 47 degrees of freedom
## Multiple R-squared:  0.963,  Adjusted R-squared:  0.962
## F-statistic:  613 on 2 and 47 DF,  p-value: <2e-16
```

```
# expend = -1.146e+02 + 2.690e-02*lawyers + 2.976e-02*employ  + error
# Step-down

summary(lm(expend~bad+crime+lawyers+employ + pop, data=regression_data))
```

```
##
## Call:
## lm(formula = expend ~ bad + crime + lawyers + employ + pop, data = regression_data)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -638.7  -92.6   23.1  117.7  792.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.14e+02   1.42e+02   -2.21   0.0322 *
## bad         -2.90e+00   1.25e+00   -2.32   0.0251 *
## crime        3.42e-02   2.84e-02    1.21   0.2345
## lawyers      2.31e-02   8.08e-03    2.86   0.0064 **
## employ       2.27e-02   7.50e-03    3.03   0.0041 **
## pop          8.06e-02   3.55e-02    2.27   0.0281 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 227 on 44 degrees of freedom
## Multiple R-squared:  0.968,  Adjusted R-squared:  0.964
## F-statistic:  264 on 5 and 44 DF,  p-value: <2e-16
```

```
summary(lm(expend~lawyers+employ+bad + pop, data=regression_data))
```

```
##
## Call:
## lm(formula = expend ~ lawyers + employ + bad + pop, data = regression_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -635.8  -79.6   19.7  116.5  799.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.52e+02   4.65e+01   -3.27   0.0020 **
## lawyers      2.65e-02   7.62e-03    3.48   0.0011 **
## employ       2.26e-02   7.54e-03    3.00   0.0044 **
## bad         -2.27e+00   1.14e+00   -1.99   0.0529 .
## pop          6.54e-02   3.33e-02    1.96   0.0560 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 228 on 45 degrees of freedom
## Multiple R-squared:  0.967,  Adjusted R-squared:  0.964
## F-statistic:  326 on 4 and 45 DF,  p-value: <2e-16
```

```
summary(lm(expend~lawyers+employ + bad, data=regression_data))
```

```
##
## Call:
## lm(formula = expend ~ lawyers + employ + bad, data = regression_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -631.8  -94.9   32.2   92.4  958.6
```

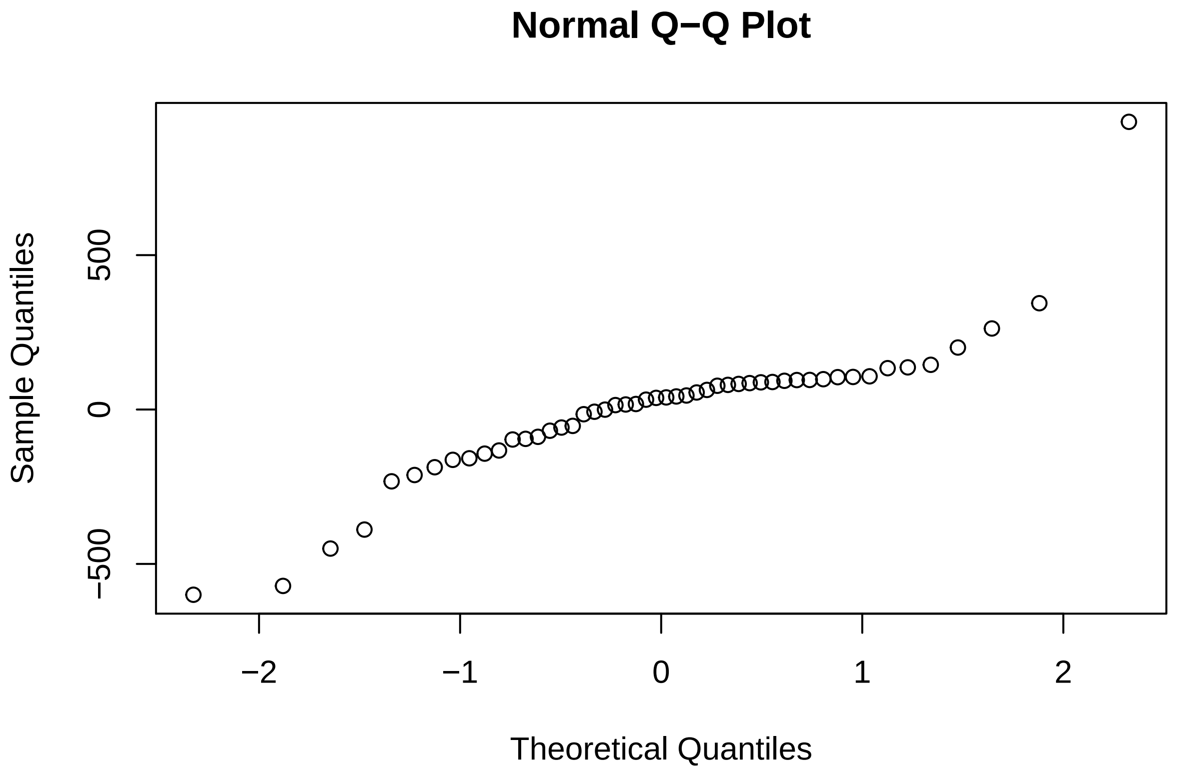
```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.14e+02   4.36e+01   -2.62   0.0118 *
## lawyers      2.63e-02   7.85e-03    3.36   0.0016 **
## employ       3.23e-02   5.85e-03    5.53  1.5e-06 ***
## bad         -8.55e-01   9.12e-01   -0.94   0.3530
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 235 on 46 degrees of freedom
## Multiple R-squared:  0.964,  Adjusted R-squared:  0.961
## F-statistic:  408 on 3 and 46 DF,  p-value: <2e-16
```

```
summary(lm(expend~lawyers+employ , data=regression_data))
```

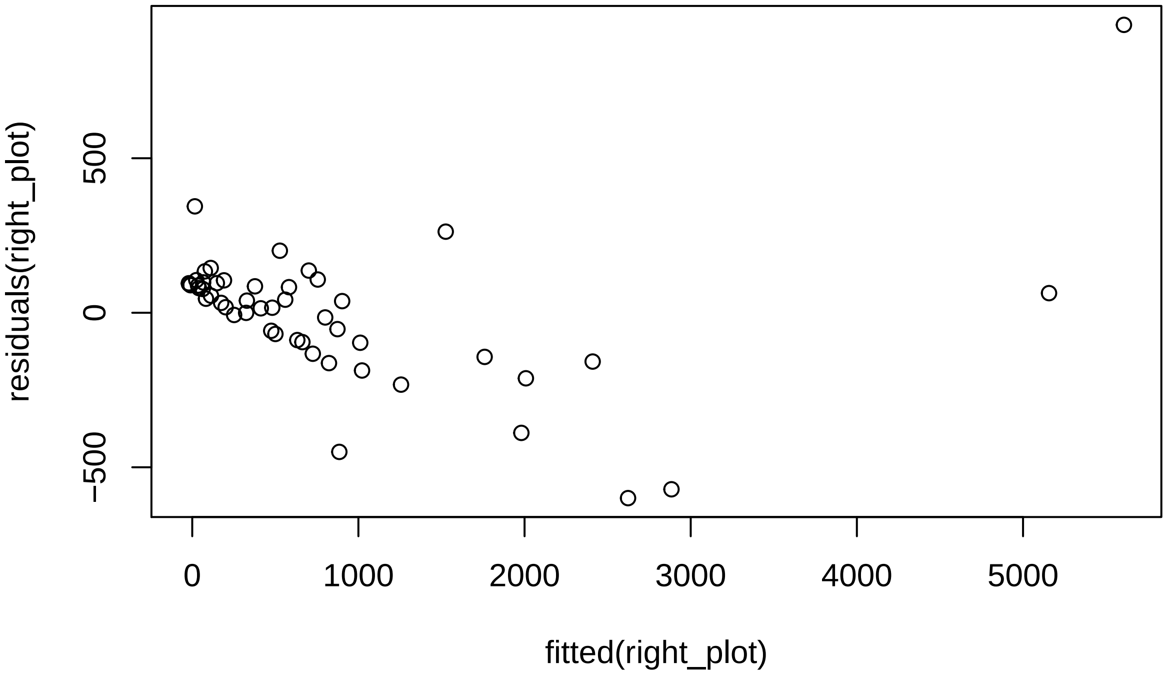
```
##
## Call:
## lm(formula = expend ~ lawyers + employ, data = regression_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -599.8  -93.4   38.4   94.8  931.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.15e+02   4.36e+01   -2.63   0.0115 *
## lawyers      2.69e-02   7.82e-03    3.44   0.0012 **
## employ       2.98e-02   5.15e-03    5.77  5.9e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 234 on 47 degrees of freedom
## Multiple R-squared:  0.963,  Adjusted R-squared:  0.962
## F-statistic:  613 on 2 and 47 DF,  p-value: <2e-16
```

c) Check the model assumptions (of the resulting model from b)) by using relevant diagnostic tools.

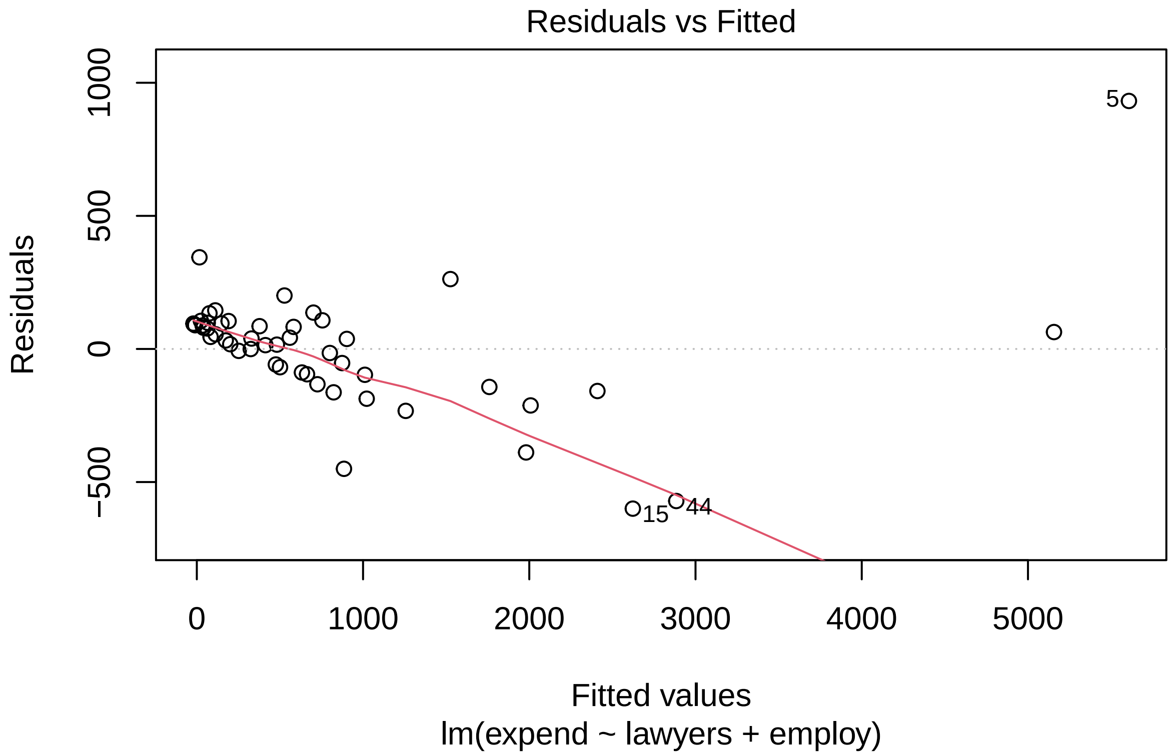
```
right_plot = lm(expend~lawyers+employ , data=regression_data)
qqnorm(residuals(right_plot))
```



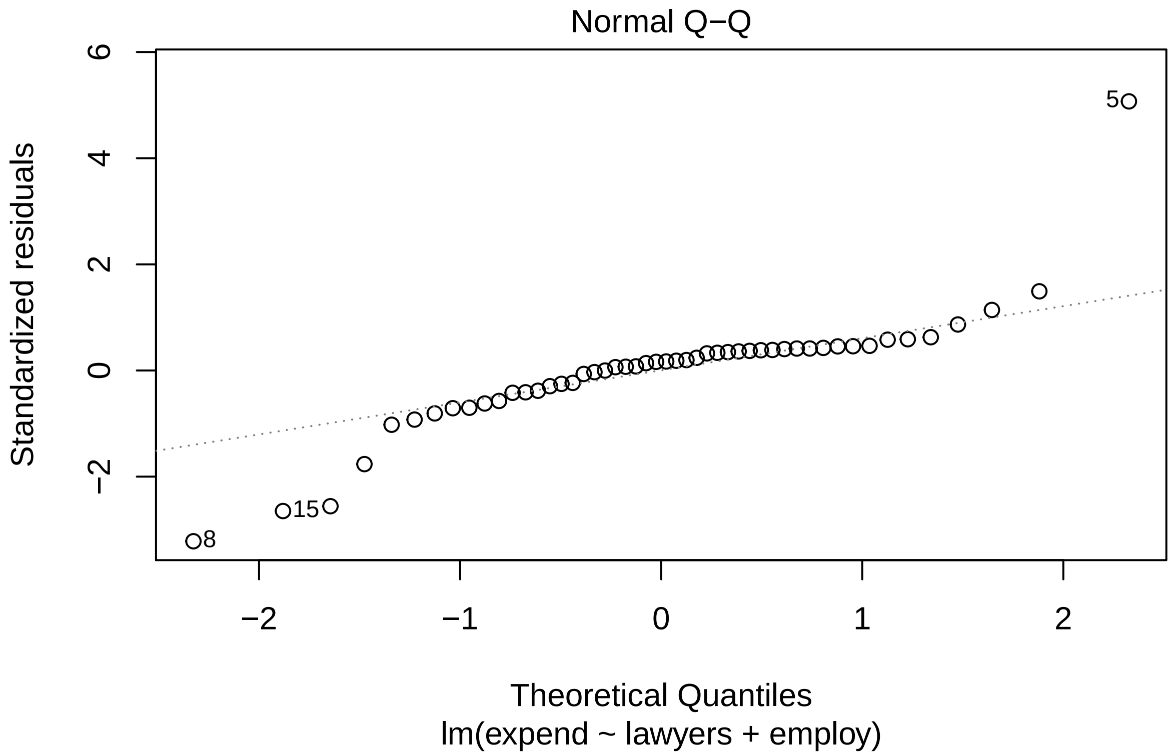
```
plot(fitted(right_plot), residuals(right_plot))
```



```
plot(right_plot, 1)
```



```
plot(right_plot, 2)
```


Normal Q−Q
Standardized residuals
6
4
2
0
−2
5
15
8
−2
−1
0
1
2
Theoretical Quantiles
lm(expend ~ lawyers + employ)