Experimental Design and Data Analysis - Assignment 2

Group 11 - Björn van der Haas, Deividas Aksomaitis, Nur Başak Özer

2023-03-16

Exercise 1: Trees

```
treedata <- read.table(file="treeVolume.txt",header=TRUE)
treedata$type <- as.factor(treedata$type)
is.factor(treedata$type); is.numeric(treedata$type)
```

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

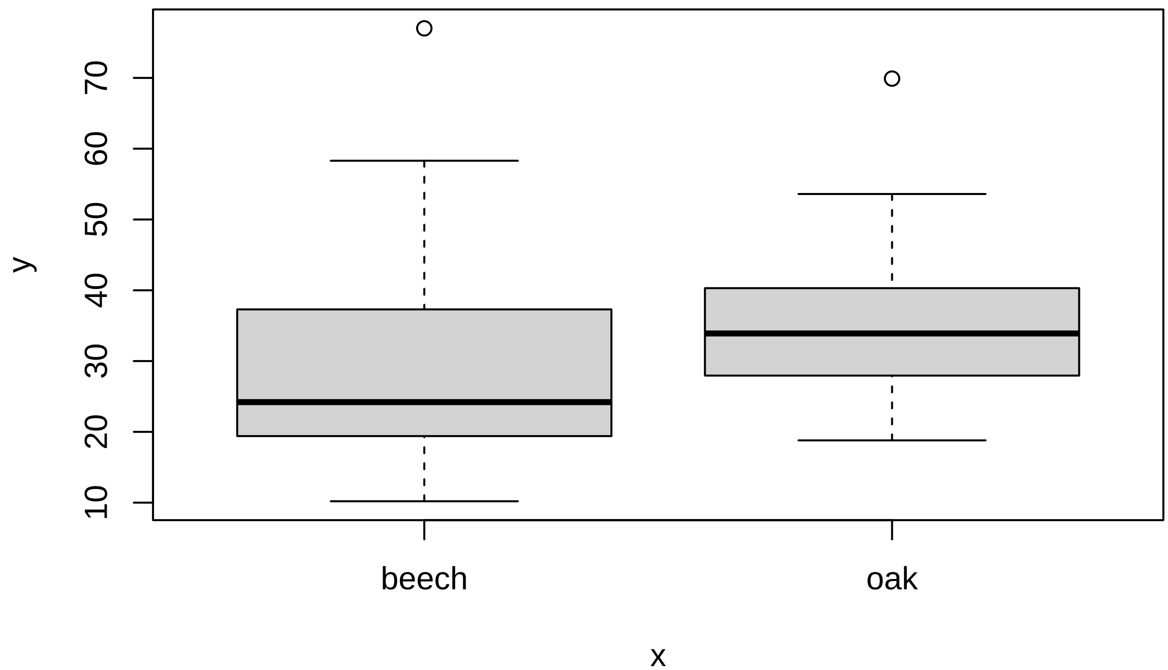
```
## [1] FALSE
```

We have an unbalanced design as there are 31 observations for beech and 27 for oak for the factor type.

Section a

As "type" is a factor, we can use a box plot to check for outliers:

```
plot(treedata$type, treedata$volume)
```



The plot indicates that each have one outlier, so we remove these 2 rows from our dataset.

```
treedata2 <- treedata[-c(31, 46),]
is.factor(treedata2$type); is.numeric(treedata2$type)
```

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

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

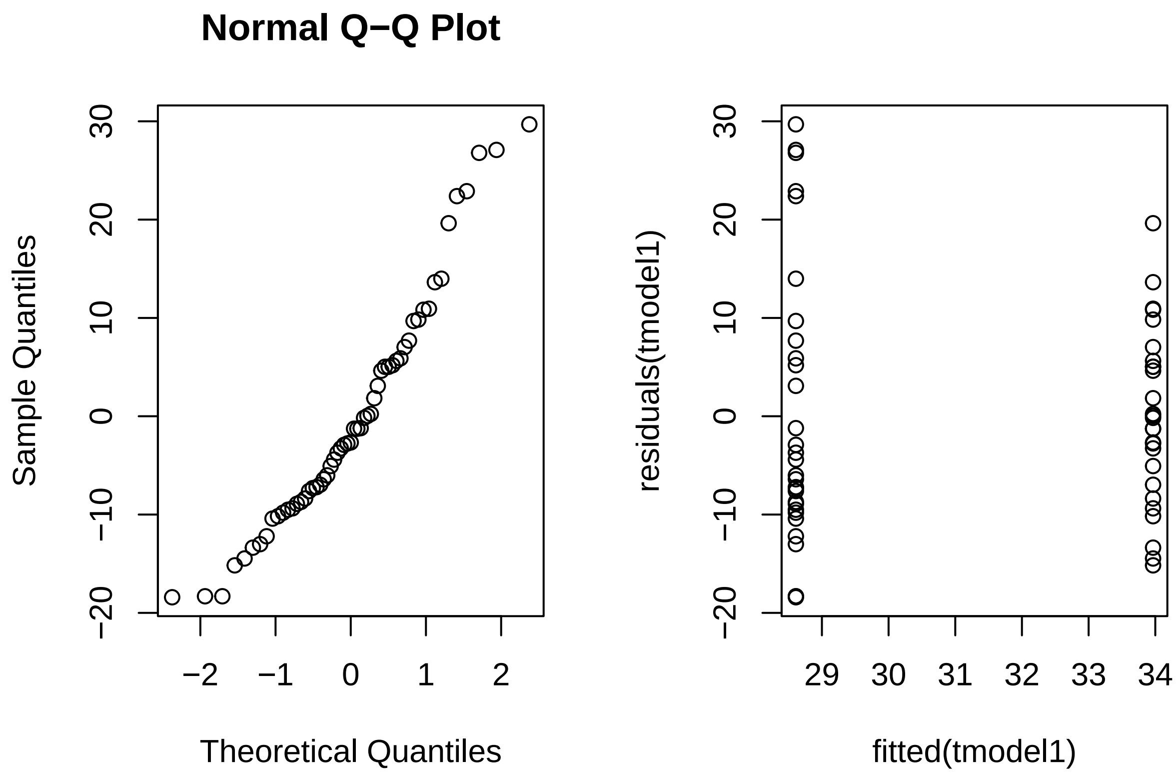
As we do not take diameter and height into account, we run a one-way ANOVA:

```
tmodel1 <- lm(volume ~ type, data=treedata2)
anova(tmodel1)
```

```
## Analysis of Variance Table
##
## Response: volume
##           Df Sum Sq Mean Sq F value  Pr(>F)
## type       1  407.8  407.76  2.8447 0.09734 .
## Residuals 55 7883.7  143.34
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As $p > 0.05$, we conclude that “type” does not have a significant effect on “volume”.

```
par(mfrow=c(1, 2))
qqnorm(residuals(tmodel1)); plot(fitted(tmodel1), residuals(tmodel1))
```



Diagnostics for ANOVA: Q-Q plot indicates doubtful normality of residuals (quite skewed), therefore the ANOVA assumptions are in question. We observe no clear relationship in Fitted vs Residuals plot, which is the desired outcome.

Because type only has 2 levels ("Oak" and "Beech"), we have a two-sample problem and as such a two-sample t-test would be sufficient:

```
t.test(treedata$volume[treedata2$type == "beech"], treedata$volume[treedata2$type == "oak"])
```

```
##
##  Welch Two Sample t-test
##
## data:  treedata$volume[treedata2$type == "beech"] and treedata$volume[treedata2$type == "oak"]
## t = -2.2974, df = 55.673, p-value = 0.02538
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -15.43060  -1.05435
## sample estimates:
## mean of x mean of y
##  28.80937  37.05185
```

Interestingly, we now have $p < 0.05$, which contradicts the ANOVA. The estimated volume for "beech" is 28.81 and for "oak" is 37.05. Had we run this test without removing outliers (code omitted but can easily be done with the original treedata) we would have a $p > 0.05$ with mean 30.17 for "beech" and the mean for "oak" would be 35.25.

The explanation for this discrepancy might be that this test requires the assumption of normality which we can test for a t-test with a Shapiro Wilk test (only reliable if it rejects normality).

```
shapiro.test(treedata$volume[treedata$type == "beech"]); shapiro.test(treedata$volume[treedata$type ==
```

```
##
##  Shapiro-Wilk normality test
##
## data:  treedata$volume[treedata$type == "beech"]
## W = 0.88757, p-value = 0.003579

##
##  Shapiro-Wilk normality test
##
## data:  treedata$volume[treedata$type == "oak"]
## W = 0.9349, p-value = 0.08199
```

$p < 0.05$ for the Shapiro-Wilk normality test for "beech", which means normality can not be assumed and a t-test is therefore not an appropriate test. (Similarly, the Shapiro-Wilk test also rejects h0 for the dataset with the removed outliers for "beech".)

We can estimate the volumes for the two tree types with the aggregate function:

```
volmean <- aggregate(volume ~ type, data = treedata2, mean)
volmean
```

```
##    type   volume
## 1 beech 28.61000
## 2   oak 33.96667
```

We obtain the results: 28.61 for "beech" and 33.97 for "oak".

Section b

We now have "volume" as the numerical outcome, the factor "type", and the numerical explanatory variable "diameter". As stated above, this is an unbalanced design. We thus run an ANCOVA with the drop1 function:

```
tancova1 = lm(volume ~ type + diameter, data = treedata2)
drop1(tancova1, test="F")
```

```
## Single term deletions
##
## Model:
## volume ~ type + diameter
##          Df Sum of Sq    RSS    AIC  F value Pr(>F)
## <none>                 773.8 154.67
## type      1      16.4  790.2 153.87   1.1424 0.2899
## diameter  1    7109.9 7883.7 284.98 496.1624 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The drop1 function allows us to interpret both p values properly. $p > 0.05$ for type, but $p < 0.05$ for diameter indicating diameter has a significant effect on volume.

We can do the same for height:

```
tancova2 = lm(volume ~ type + height, data = treedata2)
drop1(tancova2, test="F")
```

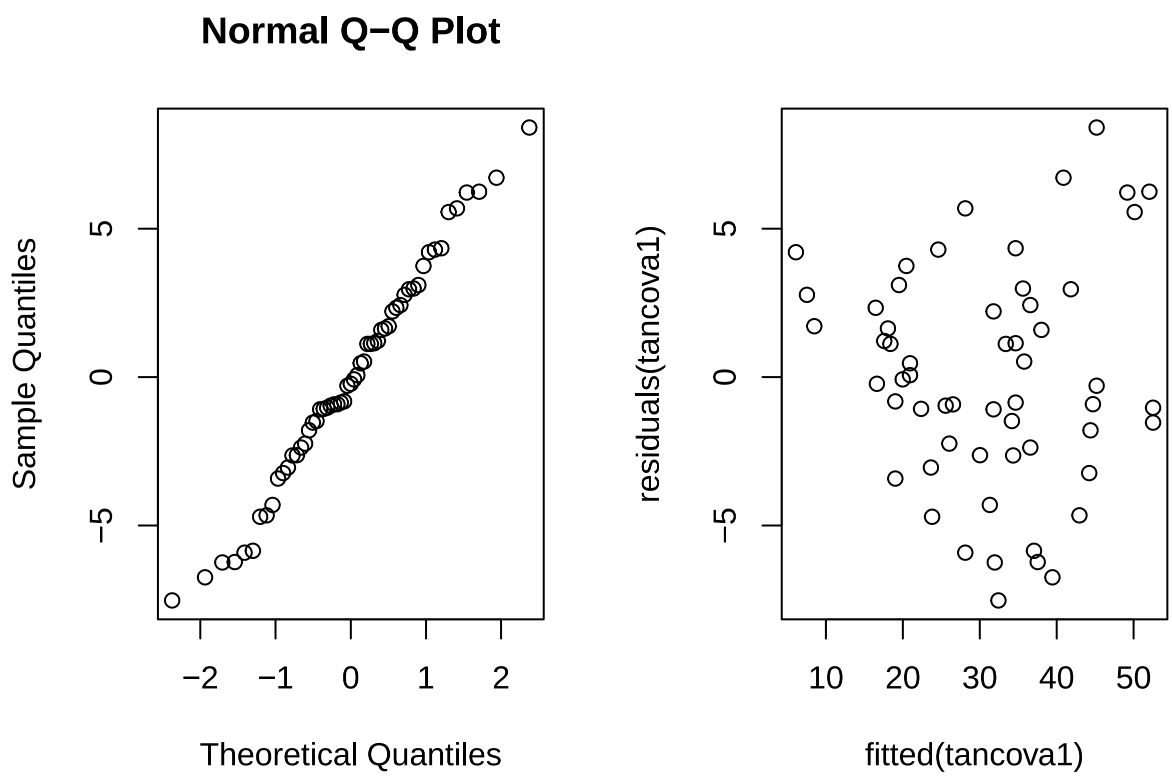
```
## Single term deletions
##
## Model:
```

```
## volume ~ type + height
##        Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>               5975.2 271.18
## type    1     358.1 6333.3 272.50  3.2363 0.0776121 .
## height  1    1908.5 7883.7 284.98 17.2473 0.0001175 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

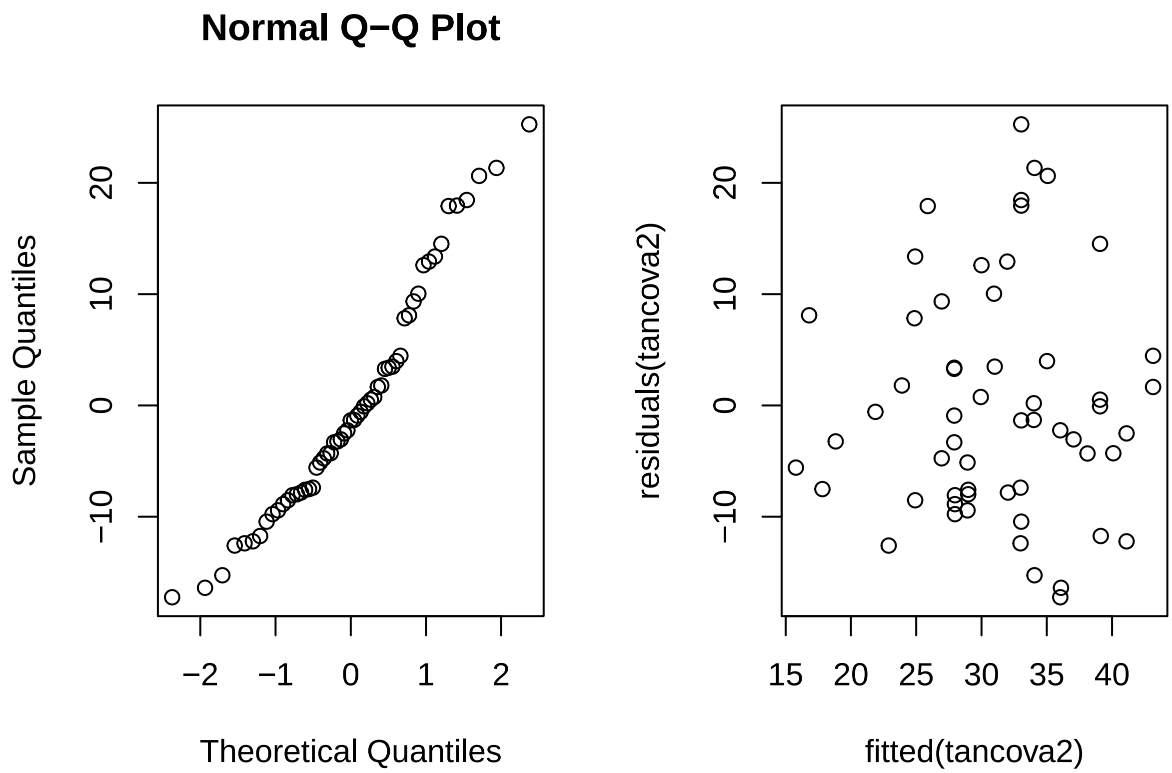
$p < 0.05$ for height, indicating it has a significant effect on volume.

Diagnostics for the two ANCOVA tests:

```r
par(mfrow=c(1, 2))
qqnorm(residuals(tancova1)); plot(fitted(tancova1), residuals(tancova1))
```




```r
par(mfrow=c(1, 2))
qqnorm(residuals(tancova2)); plot(fitted(tancova2), residuals(tancova2))
```



For the first ANCOVA, the Q-Q indicates normality and no relation between residuals and fitted as desired. For the second ANCOVA, the Q-Q is slightly skewed, but likely indicates normality, and no relation between residuals and fitted as desired.

We can also consider a pairwise interaction for “type” and “diameter” for the first model and a pairwise interaction for “type” and “height” for the second:

```
tpw1 = lm(volume ~ type*diameter, data = treedata2)
anova(tpw1)
```

```
## Analysis of Variance Table
##
## Response: volume
##               Df Sum Sq Mean Sq  F value    Pr(>F)
## type           1  407.8   407.8  27.9536 2.398e-06 ***
## diameter       1 7109.9  7109.9 487.4158 < 2.2e-16 ***
## type:diameter  1    0.7     0.7   0.0481    0.8273
## Residuals     53  773.1    14.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As $p > 0.05$, the interaction between factor “type” and predictor “diameter” does not seem to have a significant effect.

```
tpw2 = lm(volume ~ type*height, data = treedata2)
anova(tpw2)
```

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

```
## Response: volume
##              Df Sum Sq Mean Sq F value    Pr(>F)
## type          1  407.8  407.76  3.6950 0.0599592 .
## height        1 1908.4 1908.45 17.2939 0.0001177 ***
## type:height   1  126.4  126.45  1.1459 0.2892702
## Residuals    53 5848.8  110.35
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Similarly, as $p > 0.05$, the interaction between factor "type" and predictor "height" does not seem to have a significant effect. We can conclude that the influence of both diameter and height is similar for both types.

Section c

As concluded in section (b), we found no significant indicator that there was any interaction effect. We will thus analyze a purely additive model.

```
tadd = lm(volume ~ diameter + height + type, data = treedata2)
drop1(tadd, test = "F")
```

```
## Single term deletions
##
## Model:
## volume ~ diameter + height + type
##          Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                 492.6 130.92
## diameter  1    5482.7 5975.2 271.18 589.943 < 2.2e-16 ***
## height    1     281.2  773.8 154.67  30.263 1.112e-06 ***
## type      1       8.4  501.0 129.89   0.905    0.3458
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This further confirms that factor "type" is not significant, so we can continue with an additive model without this factor.

```
tadd2 = lm(volume ~ diameter + height, data = treedata2)
drop1(tadd2, test = "F")
```

```
## Single term deletions
##
## Model:
## volume ~ diameter + height
##          Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                 501.0 129.89
## diameter  1    5832.4 6333.3 272.50 628.676 < 2.2e-16 ***
## height    1     289.2  790.2 153.87  31.174 7.867e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

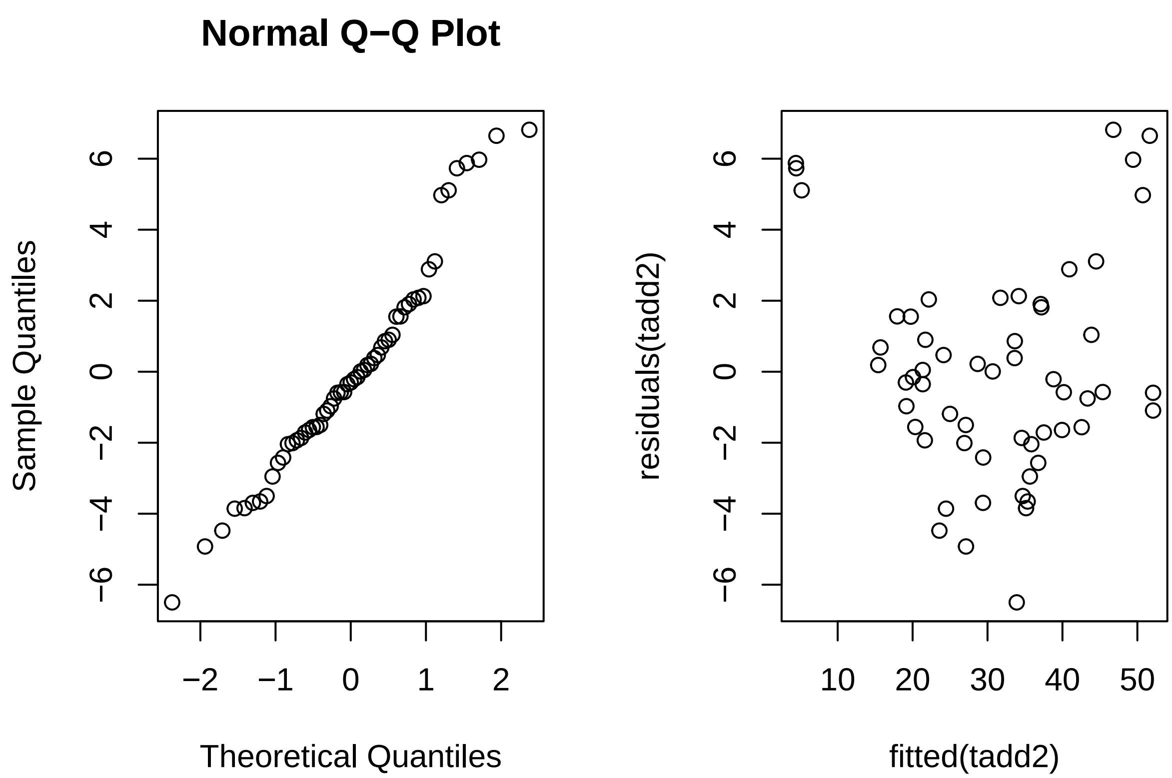
```
summary(tadd2)
```

```
##
## Call:
## lm(formula = volume ~ diameter + height, data = treedata2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -6.4979 -1.8660 -0.3036  1.5585  6.8144
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -60.61020    5.40545 -11.213 1.02e-15 ***
## diameter      4.40481    0.17568  25.073  < 2e-16 ***
## height        0.41772    0.07482   5.583 7.87e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 54 degrees of freedom
## Multiple R-squared:  0.9396, Adjusted R-squared:  0.9373
## F-statistic: 419.9 on 2 and 54 DF,  p-value: < 2.2e-16
```

Our new model further indicates significance for both diameter and height and has a very high R-squared. Notably, we now have fewer variables, making this high R-squared more relevant.

```
par(mfrow=c(1, 2))
qqnorm(residuals(tadd2)); plot(fitted(tadd2), residuals(tadd2))
```



Diagnostics mostly indicate normality in the Q-Q plot, albeit slightly skewed. Fitted vs residuals do not show a clear relation either. We believe the model assumptions to be valid.

In conclusion, we can assume that the factor "type" does not affect response value "volume" in a significant fashion. However, both explanatory variables "diameter" and "height" do have a significant impact, with diameter having the heaviest weight with 4.4 while height has 0.42.

We can now predict the overall average diameter and height with the following linear regression model:

$$volume = -60.61 + 4.4 * diameter + 0.42 * height$$

```
meand <- mean(treedata2$diameter)
meanh <- mean(treedata2$height)
overallmean <-  data.frame(diameter=c(meand), height=c(meanh))
predict(tadd2, overallmean, interval = "confidence")
```

```
##        fit      lwr     upr
## 1 31.14737 30.33853 31.9562
```

We predict the volume of the overall average tree to be 31.15 and have a 95% CI of [30.34, 31.96].

Section d

The two explanatory variables that thus far were relevant were diameter and height, therefore we can drop type as a consideration.

A possible transformation would be considering a tree as a cylindrical object. A cylinder's volume can be calculated as $V = \pi * r2 * h$ where radius squared is the same as diameter. We apply this transformation to the explanatory variables and use this for the basis of a new model:

```
treedata2$volnew <- pi * treedata2$diameter * treedata2$height
volmodel <- lm(volume ~ volnew, data = treedata2)
anova(volmodel)
```

```
## Analysis of Variance Table
##
## Response: volume
##           Df Sum Sq Mean Sq F value    Pr(>F)
## volnew     1 7758.5  7758.5  800.69 < 2.2e-16 ***
## Residuals 55  532.9     9.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

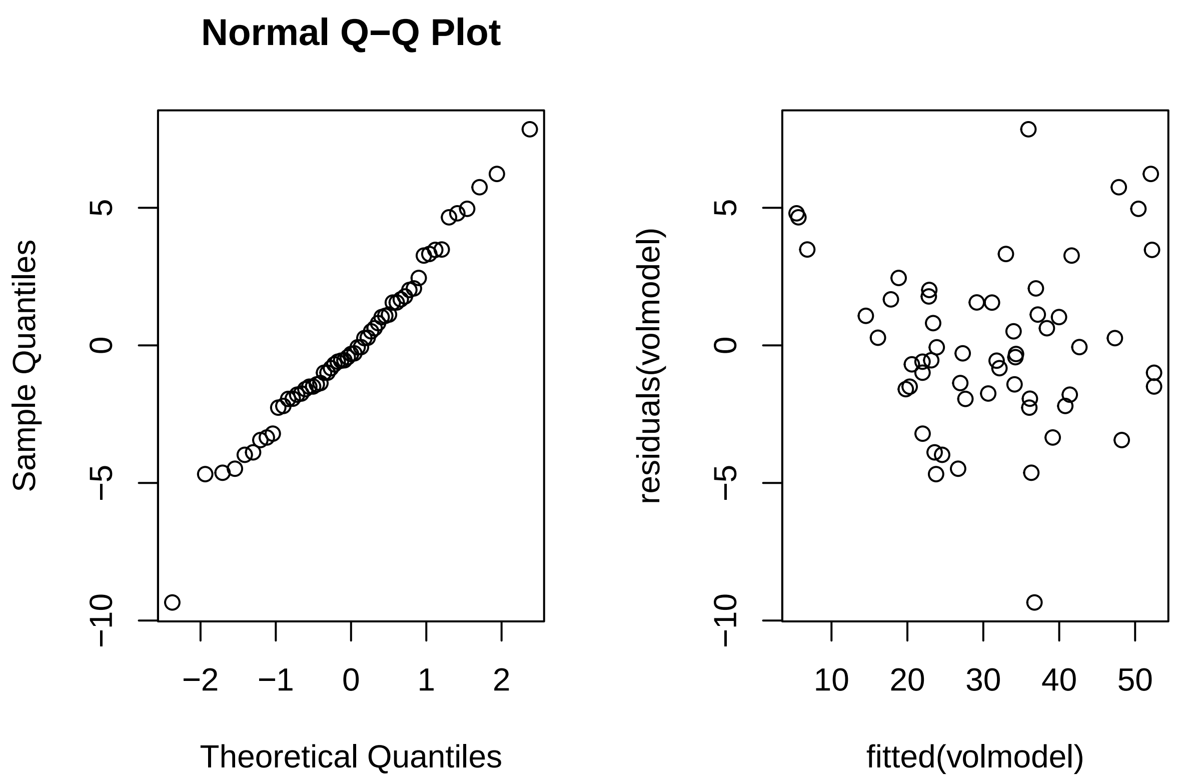
```
summary(volmodel)
```

```
##
## Call:
## lm(formula = volume ~ volnew, data = treedata2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3435 -1.7494 -0.3133  1.6717  7.8542
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.408e+01  1.995e+00  -12.07   <2e-16 ***
## volnew       1.693e-02  5.982e-04   28.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.113 on 55 degrees of freedom
## Multiple R-squared:  0.9357, Adjusted R-squared:  0.9346
## F-statistic: 800.7 on 1 and 55 DF,  p-value: < 2.2e-16
```

This new explanatory variable has a significant effect as $p < 0.05$ and an R-squared of 0.9357 which is slightly lower than previous models, but an argument can be made that by reducing to one variable it is more reliable and better explains the data.

Diagnostics:

```
par(mfrow=c(1, 2))
qqnorm(residuals(volmodel)); plot(fitted(volmodel), residuals(volmodel))
```



Q-Q plot indicates assumptions of normality of residuals holds, while no clear relation is shown in the fitted vs residuals plot.

Exercise 2: Expenditure on criminal activities

```
data <- read.table(file="expensescrime.txt",header=TRUE)

# We exclude the 1st column as it will not be part of our model.
expcr <- data[,-1]
expcr
```

```
##    expend   bad crime lawyers employ   pop
## 1     360   5.1  5877    1749   2796   525
## 2     498  34.4  3942    6679  13999  4083
## 3     219  19.2  3585    3741   7227  2388
## 4     728  31.3  7116    7535  14755  3386
## 5    6539 336.2  6518   82001 118149 27663
## 6     602  25.7  6919   11174  12556  3296
```

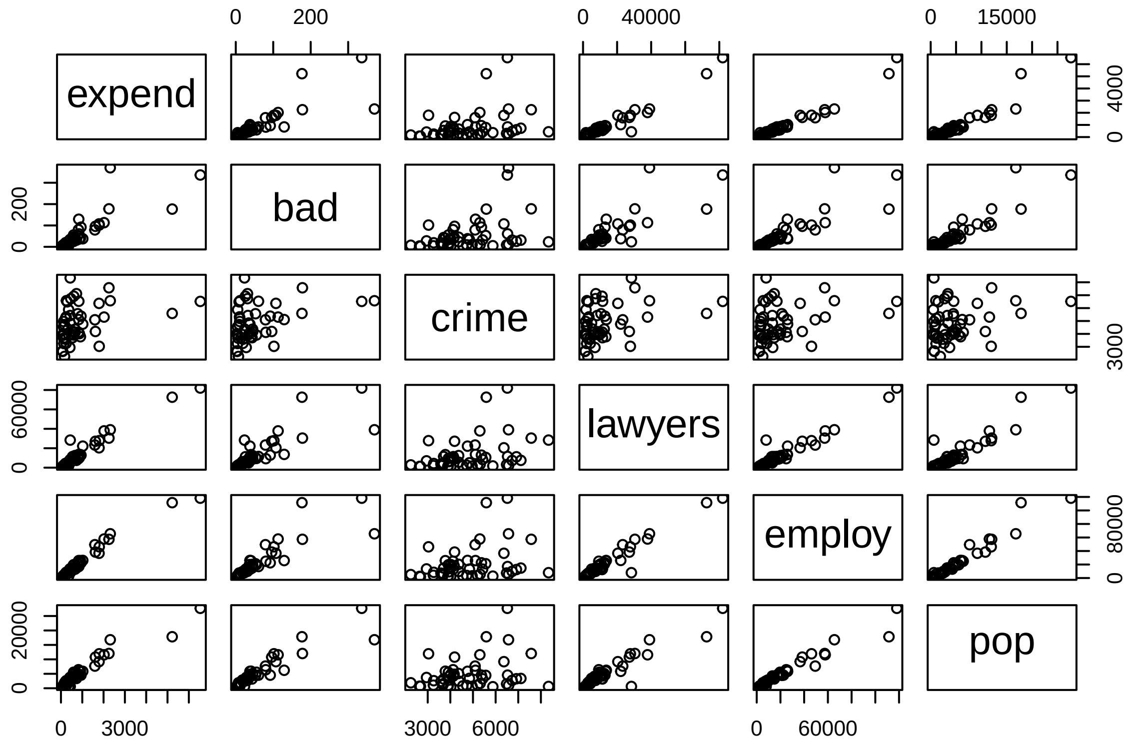
```
## 7      544  43.5  3705   11397  14798  3211
## 8      435  23.3  8339   28399   7925   622
## 9      130  10.6  4961    1597   3230   644
## 10    2252 177.9  7574   30444  57310 12023
## 11     835 129.2  5110   13652  25848  6222
## 12     210  10.8  5201    2787   3886  1083
## 13     368  17.7  3943    6182   9309  2834
## 14     120   5.8  3908    2031   3363   998
## 15    2023 113.0  5303   37873  57748 11582
## 16     593  55.3  3914    9499  19647  5531
## 17     324  23.8  4375    5555   9726  2476
## 18     417  27.9  2947    7017  13480  3727
## 19     785  52.7  5564   10569  21184  4461
## 20    1024  37.8  4758   22154  26048  5855
## 21     940  92.0  5373   12866  22541  4535
## 22     128   6.3  3672    2528   4340  1187
## 23    1788 107.2  6366   20445  36632  9200
## 24     665  38.6  4134   11343  13159  4246
## 25     660  44.9  4366   12439  20260  5103
## 26     245  18.9  3266    4270   8463  2625
## 27     123   4.9  4549    2006   3211   809
## 28     821  80.2  4121    9265  24843  6413
## 29      75   2.4  2679    1290   1997   672
## 30     206  13.7  3695    4289   5820  1594
## 31     140   4.8  3252    2139   4034  1057
## 32    1592  79.2  5094   23301  49346  7672
## 33     296   8.9  6486    3164   7413  1500
## 34     256  11.4  6575    2276   5528  1007
## 35    5220 176.7  5589   72575 111518 17825
## 36    1617  96.0  4187   27191  38404 10784
## 37     432  32.4  5425    8302  13167  3272
## 38     463  31.2  6730    7385   9858  2724
## 39    1796 101.9  3037   27798  46200 11936
## 40     164   9.2  4723    2527   3774   986
## 41     427  34.5  4841    5021  13177  3425
## 42      79   3.9  2641    1230   2396   709
## 43     568  45.2  4167    8782  18190  4855
## 44    2313 370.1  6569   39028  65488 16789
## 45     244  10.0  5317    3446   5715  1680
## 46     914  40.5  3779   13390  25720  5904
## 47      74   6.2  3888    1372   1969   548
## 48     838  60.7  6529   11507  17020  4538
## 49     863  36.6  4017   10316  19911  4807
## 50     168   7.2  2253    2835   5079  1897
## 51     115   3.1  4015    1116   2558   490
```

```
expcrlm = lm(expend ~ bad + crime + lawyers + employ + pop, data=expcr)
```

Section a

For this task, we need to make some graphical summaries of the data. Since we also need to investigate the problem of influence points and collinearity, we can utilize a number of graphical diagnostic tools here. One such tool we can use to check model quality is scatter plot, which can be used to observe linear relationships between explanatory variables:

```r
pairs(expcr)
```



Based on this scatter plot, we can observe linear relationships between explanatory variables: *bad* and *lawyers*, *bad* and *employ*, *bad* and *pop*, *lawyers* and *employ*, *lawyers* and *pop*, *employ* and *pop*.

Collinearity is the problem of linear relations between **explanatory variables.** Hence we do not mention any linear relationships between the response variable, *expend*. On the other hand, we include every single pair that corresponds to a straight line in a scatter plot as they carry the same information. Based on the above list, we can conclude that our model most definitely suffers from the collinearity problem.

To really make sure that we are dealing with this problem, we should also compute the variance inflation factors (VIF) of the explanatory variables:

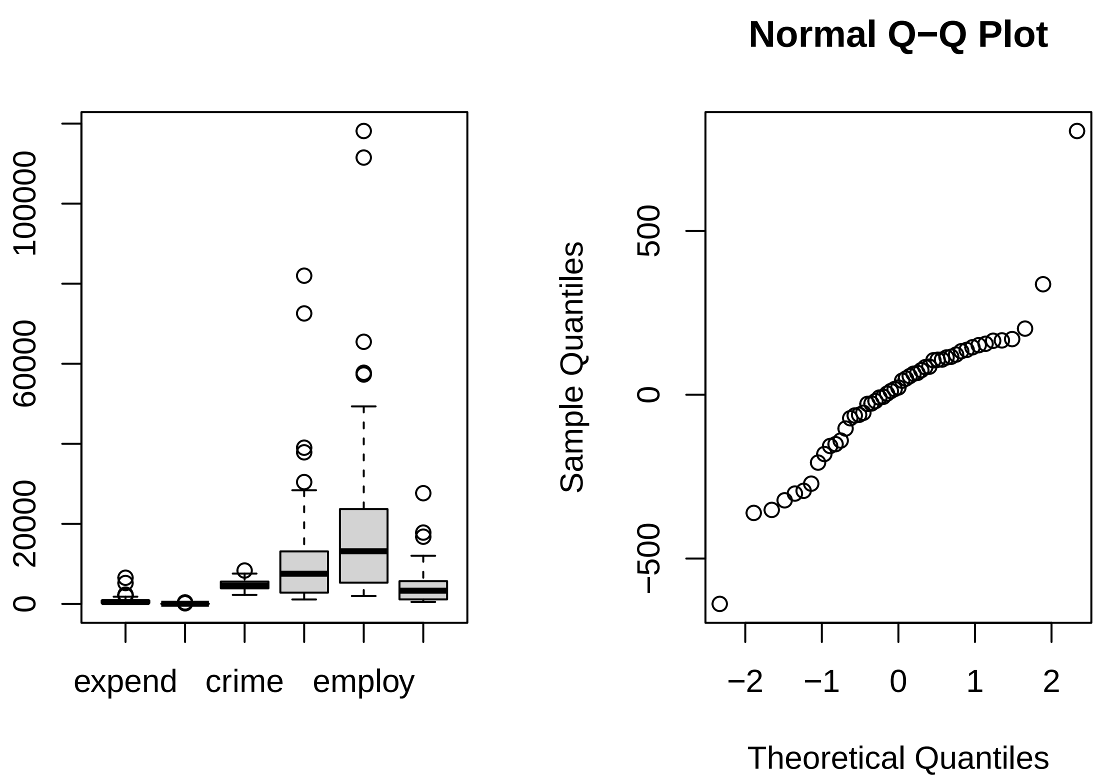
```r
library(car); vif(expcrlm)
```

```
## Zorunlu paket yükleniyor: carData

##       bad     crime   lawyers    employ       pop
##  8.364321  1.487978 16.967470 33.591361 32.937517
```

Rule of thumb suggests that if the VIF of an explanatory variable is larger than 5, then that variable is a linear combination of other variables. Here, we see that except the variable *crime*, all the other variables have VIF values larger than 5. Notice that *crime* is also not part of the above reported pair of variables.

In addition to the above graphical summary, we might want to identify the outlying values on a closer look. To do that, we can take a look at the box plot of the data or the QQ-plot of the residuals:

```r
par(mfrow=c(1,2))

boxplot(expcr); qqnorm(residuals(expcrlm))
```



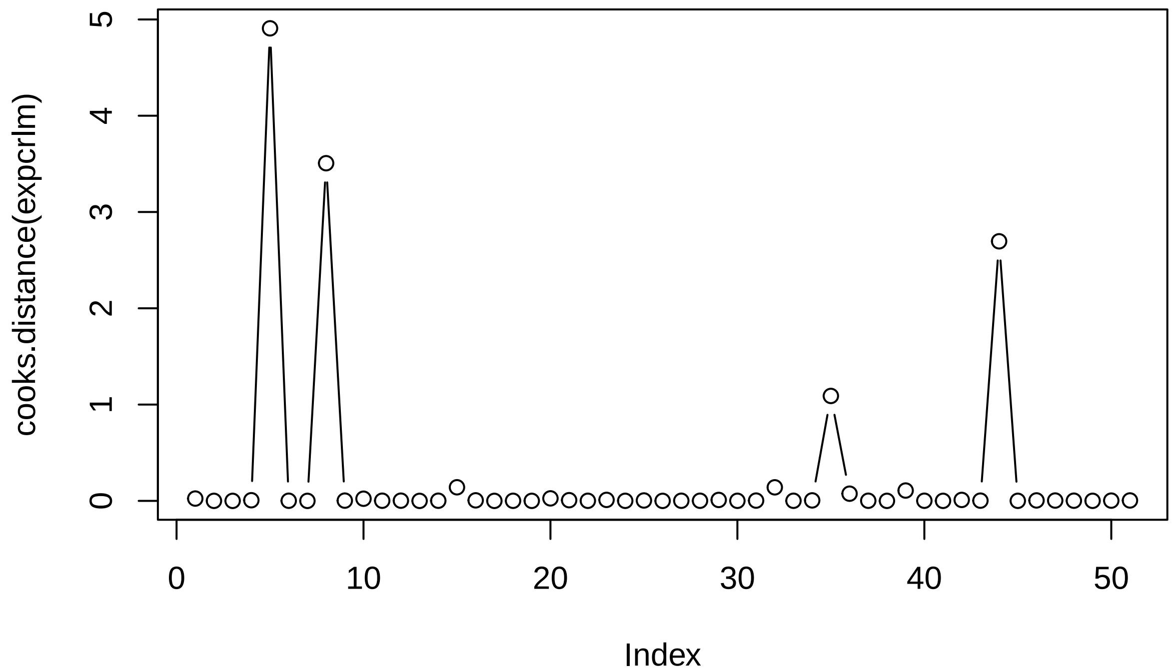
It looks there are quite many outliers outside most of the box plots, hence the data may be suffering from inconsistency. Moreover, from the above QQ-plot we can identify quite a number of outliers e.g. the data point that is furthest to the right appears to behave vastly different than the others.

Let us now study the effect of the influence points in our data. To do that, we must compute and plot the Cook's distances of the data points in the model:

```
round(cooks.distance(expcrlm),3)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## 0.024 0.000 0.000 0.007 4.908 0.001 0.000 3.507 0.003 0.022 0.002 0.002 0.000
##    14    15    16    17    18    19    20    21    22    23    24    25    26
## 0.001 0.141 0.005 0.000 0.001 0.000 0.026 0.007 0.001 0.010 0.000 0.004 0.000
##    27    28    29    30    31    32    33    34    35    36    37    38    39
## 0.001 0.001 0.010 0.001 0.003 0.140 0.001 0.005 1.090 0.075 0.001 0.000 0.107
##    40    41    42    43    44    45    46    47    48    49    50    51
## 0.001 0.000 0.010 0.002 2.696 0.000 0.005 0.003 0.002 0.001 0.003 0.004
```

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



Rule of thumb suggests that if the Cook's distance of a data point is larger than 1, it shall be considered an influence point. Thus, based on the above computations, we conclude that data points with the indexes 5,8,35,44 are influence points.

Section b

In this section, we need to use the step-up method to come up with a model with the best choice of explanatory variables. However, in order to perform a better statistical analysis overall, we could remove the influence points we identified in the previous section from the data set:

```
new_expcr <- expcr[-c(5, 8, 35, 44), ]
new_expcr
```

```
##    expend   bad crime lawyers employ   pop
## 1     360   5.1  5877    1749   2796   525
## 2     498  34.4  3942    6679  13999  4083
## 3     219  19.2  3585    3741   7227  2388
## 4     728  31.3  7116    7535  14755  3386
## 6     602  25.7  6919   11174  12556  3296
## 7     544  43.5  3705   11397  14798  3211
## 9     130  10.6  4961    1597   3230   644
## 10   2252 177.9  7574   30444  57310 12023
## 11    835 129.2  5110   13652  25848  6222
## 12    210  10.8  5201    2787   3886  1083
## 13    368  17.7  3943    6182   9309  2834
## 14    120   5.8  3908    2031   3363   998
## 15   2023 113.0  5303   37873  57748 11582
```

```
## 16    593  55.3  3914   9499  19647  5531
## 17    324  23.8  4375   5555   9726  2476
## 18    417  27.9  2947   7017  13480  3727
## 19    785  52.7  5564  10569  21184  4461
## 20   1024  37.8  4758  22154  26048  5855
## 21    940  92.0  5373  12866  22541  4535
## 22    128   6.3  3672   2528   4340  1187
## 23   1788 107.2  6366  20445  36632  9200
## 24    665  38.6  4134  11343  13159  4246
## 25    660  44.9  4366  12439  20260  5103
## 26    245  18.9  3266   4270   8463  2625
## 27    123   4.9  4549   2006   3211   809
## 28    821  80.2  4121   9265  24843  6413
## 29     75   2.4  2679   1290   1997   672
## 30    206  13.7  3695   4289   5820  1594
## 31    140   4.8  3252   2139   4034  1057
## 32   1592  79.2  5094  23301  49346  7672
## 33    296   8.9  6486   3164   7413  1500
## 34    256  11.4  6575   2276   5528  1007
## 36   1617  96.0  4187  27191  38404 10784
## 37    432  32.4  5425   8302  13167  3272
## 38    463  31.2  6730   7385   9858  2724
## 39   1796 101.9  3037  27798  46200 11936
## 40    164   9.2  4723   2527   3774   986
## 41    427  34.5  4841   5021  13177  3425
## 42     79   3.9  2641   1230   2396   709
## 43    568  45.2  4167   8782  18190  4855
## 45    244  10.0  5317   3446   5715  1680
## 46    914  40.5  3779  13390  25720  5904
## 47     74   6.2  3888   1372   1969   548
## 48    838  60.7  6529  11507  17020  4538
## 49    863  36.6  4017  10316  19911  4807
## 50    168   7.2  2253   2835   5079  1897
## 51    115   3.1  4015   1116   2558   490
```

Now we can proceed with the step-up strategy. To do that, we must start with a background model and work our ways towards the full model by adding one new variable that yields the maximum increase in R^2 compared to other potential variables.

```
summary(lm(expend ~ bad, data=new_expcr))
```

Adding the first variable:

```
##
## Call:
## lm(formula = expend ~ bad, data = new_expcr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -919.19  -97.16  -43.96   69.03  475.09
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 109.0511    49.3073   2.212   0.0321 *
```

```
## bad          12.7332     0.8901  14.306   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 237.4 on 45 degrees of freedom
## Multiple R-squared:  0.8198, Adjusted R-squared:  0.8158
## F-statistic: 204.7 on 1 and 45 DF,  p-value: < 2.2e-16
```

```
summary(lm(expend ~ crime, data=new_expcr))
```

```
##
## Call:
## lm(formula = expend ~ crime, data = new_expcr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -646.4 -367.0 -160.6  140.8 1424.8
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -84.75256  290.60432  -0.292   0.7719
## crime         0.15014    0.06048   2.483   0.0168 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 524.4 on 45 degrees of freedom
## Multiple R-squared:  0.1205, Adjusted R-squared:  0.1009
## F-statistic: 6.163 on 1 and 45 DF,  p-value: 0.01684
```

```
summary(lm(expend ~ lawyers, data=new_expcr))
```

```
##
## Call:
## lm(formula = expend ~ lawyers, data = new_expcr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -378.71  -57.80  -33.47   66.68  490.24
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.293454  32.178322   1.314    0.195
## lawyers      0.061407   0.002548  24.102   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 149.9 on 45 degrees of freedom
## Multiple R-squared:  0.9281, Adjusted R-squared:  0.9265
## F-statistic: 580.9 on 1 and 45 DF,  p-value: < 2.2e-16
```

```
summary(lm(expend ~ employ, data=new_expcr))
```

```
##
## Call:
## lm(formula = expend ~ employ, data = new_expcr)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -263.83  -65.84  -18.53   51.71  405.38
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.202797  25.615336    0.75    0.457
## employ       0.037219   0.001195   31.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 117.7 on 45 degrees of freedom
## Multiple R-squared:  0.9557, Adjusted R-squared:  0.9547
## F-statistic: 970.2 on 1 and 45 DF,  p-value: < 2.2e-16
```

```
summary(lm(expend ~ pop, data=new_expcr))
```

```
##
## Call:
## lm(formula = expend ~ pop, data = new_expcr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -303.59 -114.69   -1.49   76.24  334.05
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.929886  33.971356  -1.087    0.283
## pop           0.168780   0.006859  24.607   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 147.1 on 45 degrees of freedom
## Multiple R-squared:  0.9308, Adjusted R-squared:  0.9293
## F-statistic: 605.5 on 1 and 45 DF,  p-value: < 2.2e-16
```

According to the above summaries of 5 different models, the addition of variable *employ* would yield the maximum increase in R^2 compared to the other variables. Moreover, we observe that the p-value reserved for *employ* is significantly smaller than 0.05. Hence we decide to include ***employ*** in the model as our first variable.

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

Adding another variable:

```
##
## Call:
## lm(formula = expend ~ employ + bad, data = new_expcr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -245.92  -63.64  -19.95   44.20  379.60
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 18.578981  25.287705   0.735    0.466
## employ       0.033478   0.002792  11.992 1.85e-15 ***
## bad          1.524846   1.031229   1.479    0.146
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 116.2 on 44 degrees of freedom
## Multiple R-squared:  0.9578, Adjusted R-squared:  0.9559
## F-statistic:   499 on 2 and 44 DF,  p-value: < 2.2e-16
```

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

```
##
## Call:
## lm(formula = expend ~ employ + crime, data = new_expcr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -250.60  -54.77  -11.41   41.68  351.20
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.651e+02  5.872e+01  -2.811  0.00735 **
## employ       3.623e-02  1.113e-03  32.557  < 2e-16 ***
## crime        4.313e-02  1.264e-02   3.411  0.00140 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 105.9 on 44 degrees of freedom
## Multiple R-squared:  0.9649, Adjusted R-squared:  0.9634
## F-statistic: 605.6 on 2 and 44 DF,  p-value: < 2.2e-16
```

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

```
##
## Call:
## lm(formula = expend ~ employ + lawyers, data = new_expcr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -226.56  -57.21  -19.45   41.96  419.51
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.626415  24.280210   0.726   0.4717
## employ       0.026751   0.004381   6.106 2.35e-07 ***
## lawyers      0.018143   0.007334   2.474   0.0173 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 111.5 on 44 degrees of freedom
## Multiple R-squared:  0.9611, Adjusted R-squared:  0.9593
## F-statistic: 543.3 on 2 and 44 DF,  p-value: < 2.2e-16
```

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

```
##
## Call:
## lm(formula = expend ~ employ + pop, data = new_expcr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -202.59  -76.13   -6.78   57.82  355.90
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.097230  26.488968  -0.192   0.8483
## employ       0.026662   0.004616   5.776  7.2e-07 ***
## pop          0.050057   0.021210   2.360   0.0228 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 112.2 on 44 degrees of freedom
## Multiple R-squared:  0.9607, Adjusted R-squared:  0.9589
## F-statistic: 537.1 on 2 and 44 DF,  p-value: < 2.2e-16
```

According to the above summaries of 4 new models, the addition of variable *crime* would yield the maximum increase in R^2 compared to the other variables. Moreover, we observe that the p-value reserved for *crime* is smaller than 0.05. Hence we decide to include ***crime*** in the model as our next variable.

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

Adding another variable:

```
##
## Call:
## lm(formula = expend ~ employ + crime + bad, data = new_expcr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -217.90  -53.57  -10.09   35.10  341.06
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.543e+02  6.041e+01  -2.554  0.01426 *
## employ       3.435e-02  2.569e-03  13.371  < 2e-16 ***
## crime        4.054e-02  1.309e-02   3.098  0.00343 **
## bad          7.921e-01  9.724e-01   0.815  0.41979
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 106.3 on 43 degrees of freedom
## Multiple R-squared:  0.9655, Adjusted R-squared:  0.9631
## F-statistic: 400.8 on 3 and 43 DF,  p-value: < 2.2e-16
```

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

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

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -213.39  -47.00  -16.04   49.06  365.48
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.657e+02  5.472e+01  -3.028 0.004153 **
## employ       2.586e-02  3.882e-03   6.663 3.97e-08 ***
## crime        4.291e-02  1.178e-02   3.642 0.000722 ***
## lawyers      1.798e-02  6.486e-03   2.772 0.008194 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 98.64 on 43 degrees of freedom
## Multiple R-squared:  0.9703, Adjusted R-squared:  0.9682
## F-statistic: 467.6 on 3 and 43 DF,  p-value: < 2.2e-16
```

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

```
##
## Call:
## lm(formula = expend ~ employ + crime + pop, data = new_expcr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -179.986  -49.637    0.484   51.189  266.632
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.474e+02  5.474e+01  -4.519 4.80e-05 ***
## employ       2.093e-02  3.947e-03   5.302 3.74e-06 ***
## crime        5.430e-02  1.127e-02   4.817 1.84e-05 ***
## pop          7.136e-02  1.785e-02   3.998 0.000246 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 91.44 on 43 degrees of freedom
## Multiple R-squared:  0.9744, Adjusted R-squared:  0.9727
## F-statistic: 546.5 on 3 and 43 DF,  p-value: < 2.2e-16
```

According to the above summaries of 3 new models, the addition of variable *pop* would yield the maximum increase in R^2 compared to the other variables. Moreover, we observe that the p-value reserved for *pop* is significantly smaller than 0.05. Hence we decide to include ***pop*** in the model as our next variable.

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

Adding another variable:

```
##
## Call:
## lm(formula = expend ~ employ + crime + pop + bad, data = new_expcr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -176.244  -49.021    0.592   52.099  268.413
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.608e+02  5.900e+01  -4.421 6.81e-05 ***
## employ       2.129e-02  4.014e-03   5.303 3.96e-06 ***
## crime        5.696e-02  1.209e-02   4.711 2.71e-05 ***
## pop          7.614e-02  1.947e-02   3.910 0.000331 ***
## bad         -5.822e-01  9.128e-01  -0.638 0.527050
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 92.08 on 42 degrees of freedom
## Multiple R-squared:  0.9747, Adjusted R-squared:  0.9723
## F-statistic: 404.3 on 4 and 42 DF,  p-value: < 2.2e-16
```

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

```
##
## Call:
## lm(formula = expend ~ employ + crime + pop + lawyers, data = new_expcr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -160.512  -43.428   -1.281   51.329  288.502
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.357e+02  5.338e+01  -4.416 6.92e-05 ***
## employ       1.632e-02  4.497e-03   3.629 0.000766 ***
## crime        5.251e-02  1.096e-02   4.791 2.09e-05 ***
## pop          6.087e-02  1.811e-02   3.361 0.001662 **
## lawyers      1.189e-02  6.101e-03   1.949 0.058003 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 88.6 on 42 degrees of freedom
## Multiple R-squared:  0.9766, Adjusted R-squared:  0.9743
## F-statistic: 437.5 on 4 and 42 DF,  p-value: < 2.2e-16
```

According to the above summaries of 2 new models, the addition of variable *lawyers* would yield the maximum increase in R^2 compared to the other variables. However, we observe that the p-value reserved for *lawyers* is slightly larger than 0.05. Hence we decide to **not** include ***lawyers*** in the model as our next variable.

At first glance, our resulting model would have 3 explanatory variables: *employ*, *crime* and *pop*. However, based on our findings from Section a, the variables *employ* and *pop* exhibit a linear relationship. Hence, in order to eliminate the collinearity problem, we must remove one of them from the model. Let us remove the last added variable *pop* from the model (Note that this is an ad-hoc choice.). Now we can compute the variance inflation factors (VIF) of the resulting variables to see if the problem is resolved:

```
new_expcrlm = lm(expend ~ employ + crime, data=new_expcr)
vif(new_expcrlm)
```

```
##   employ    crime
## 1.072456 1.072456
```

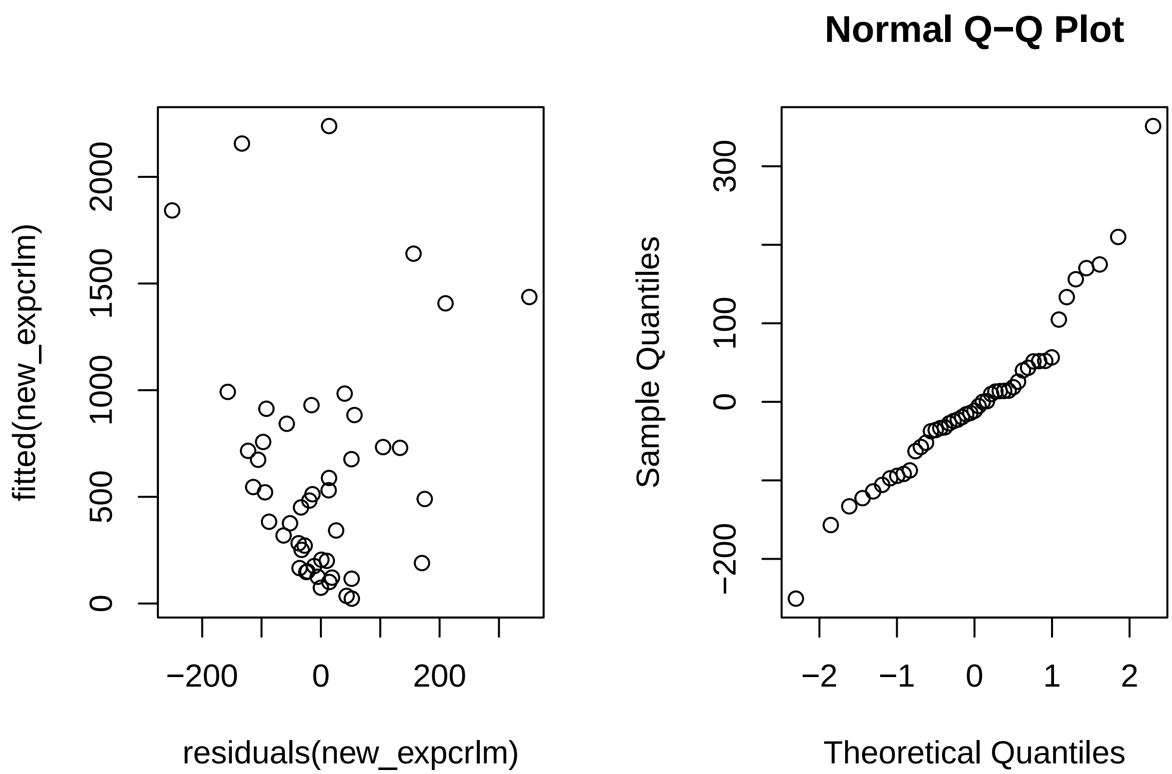
Rule of thumb suggests that if the VIF of an explanatory variable is larger than 5, then that variable is a linear combination of other variables. Here, we see that all the variables have VIF values smaller than 5.

Thus our the resulting model would be:

expend = -1.651e+02+ 3.623e-02*employ + 4.313e-02*crime + error

We can now check the model assumption for the resulting model:

```
par(mfrow=c(1,2))

plot(residuals(new_expcrlm),fitted(new_expcrlm)); qqnorm(residuals(new_expcrlm))
```



We see that the scatter plot of residuals against the observed and fitted values is all over the place as intended. Furthermore we can infer that the QQ-plot has vastly improved compared to the model plotted in Section a: the plot now resembles a straight line much more, despite a few number of existing outliers.

Section c

For this section of the exercise, we need to construct a 95% prediction interval for our new model. The x-values given for this task would be bad=50, crime=5000, lawyers=5000, employ=5000 and pop=5000, which we should use the necessary ones out of them:

```
newxdata=data.frame(crime=5000, employ=5000)
predict(new_expcrlm,newxdata,interval="prediction")
```

```
##       fit      lwr      upr
## 1 231.775 14.29633 449.2536
```

```
# Note that default significance level of interval is 0.95.
```

- **Can we improve this interval?** We know that a narrower interval is considered an improvement. Since the prediction interval is known to be always larger than the confidence interval, we can switch to constructing a confidence interval rather than a prediction interval. Moreover, we can also lower the significance level of the interval, which would result in a narrower interval.

Section d

In this section, we will apply the LASSO method to choose the relevant variables for our model. To do that, we need to install the R-package glmnet first. Then, we proceed to define the predictors and the response variable for our model:

```
library(glmnet)
```

```
## Loaded glmnet 4.1-6
```

```
x=as.matrix(new_expcr[,-1]) # remove response variable = expend
y=new_expcr[,1] # only response variable = expend
```

We then reserve 2/3 of the rows for the train set:

```
train=sample(1:nrow(x),0.67*nrow(x)) # train by using 2/3 of the x rows
x.train=x[train,]; y.train=y[train]  # data to train
x.test=x[-train,]; y.test = y[-train] # data to test the prediction quality
```

For the next step, we perform cross-validation to choose the lambda value:

```
lasso.model=glmnet(x.train,y.train,alpha=1) # alpha=1 for lasso
lasso.cv=cv.glmnet(x.train,y.train,alpha=1,type.measure="mse")
lambda.1se=lasso.cv$lambda.1se; lambda.1se
```

```
## [1] 47.01698
```

```
coef(lasso.model,s=lasso.cv$lambda.1se)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) 52.59458287
## bad          4.24370547
## crime        .
## lawyers      0.01220444
## employ       0.01193846
## pop          0.02813266
```

Judging by the results above, we can observe the explanatory variable *crime* might get disappeared due to the penalization of the model complexity. Note that, LASSO method might keep collinear variables, such as *lawyers* and *employ*, in the model like it does here.

Finally, we obtain the mean squared error value for the predicted test rows by doing the following:

```
lasso.pred=predict(lasso.model,s=lambda.1se,newx=as.matrix(x.test))
mse.lasso=mean((y.test-lasso.pred)^2); mse.lasso
```

```
## [1] 22603.11
```

We can now compare the resulting model with the model we obtained in Section b:

```
# Prediction by using the linear model

lm.model=lm(expend ~ employ + crime,data=new_expcr,subset=train) # fit linear model on the train data
y.predict.lm=predict(lm.model,newdata=new_expcr[-train,]) # predict for the test rows
mse.lm=mean((y.test-y.predict.lm)^2); mse.lm # prediction quality by the linear mode
```

```
## [1] 8326.932
```

Although we know that a new run delivers a new model because of a new train set, the model we obtained in Section b might still outperform the one we applied LASSO method on. This is because in the beginning, we had few explanatory variables to use in the construction of our model. Furthermore, the step-up method ended up constructing a much simpler model (even before addressing collinearity). Thus, to observe that LASSO method can outperform the step-down and step-up approaches, we must perform this comparative analysis over a data set with too many explanatory variables.

Exercise 3: Titanic

Section a

```
data <- read.table(file="titanic.txt",header=TRUE)
summary(data)
```

```
##      Name              PClass                Age             Sex
##  Length:1313        Length:1313        Min.   : 0.17   Length:1313
##  Class :character   Class :character   1st Qu.:21.00   Class :character
##  Mode  :character   Mode  :character   Median :28.00   Mode  :character
##                                        Mean   :30.40
##                                        3rd Qu.:39.00
##                                        Max.   :71.00
##                                        NA's   :557
##     Survived
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3427
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

We check the data with summary function to see what could be of interest to display in graphics or tables. We see that there are 557 missing 'Age' values.The most important factor here is 'Survived'; the mean of 0.34 would mean that around third of the passengers survived, however, this list is not complete as there are only 1313 entries out of 2224 total passengers.

```
tab1 <- table(data$Survived, data$PClass)
tab2 <- table(data$Survived, data$Sex)

par(mfrow=c(2,2))

barplot(tab1, beside = TRUE, main="Survival by Passenger Class",
        xlab="Passenger Class", ylab="Count",
        col=c("red", "green"), ylim=c(0,700))

barplot(tab2, beside = TRUE, main="Survival by Sex",
        xlab="Passenger Sex", ylab="Count",
        col=c("red", "green"), ylim=c(0,800))

hist(data$Age[data$Survived==1], beside = TRUE, main="Survival by Age",
        xlab="Passenger Age", ylab="Count", col="green", ylim=c(0,60))
```

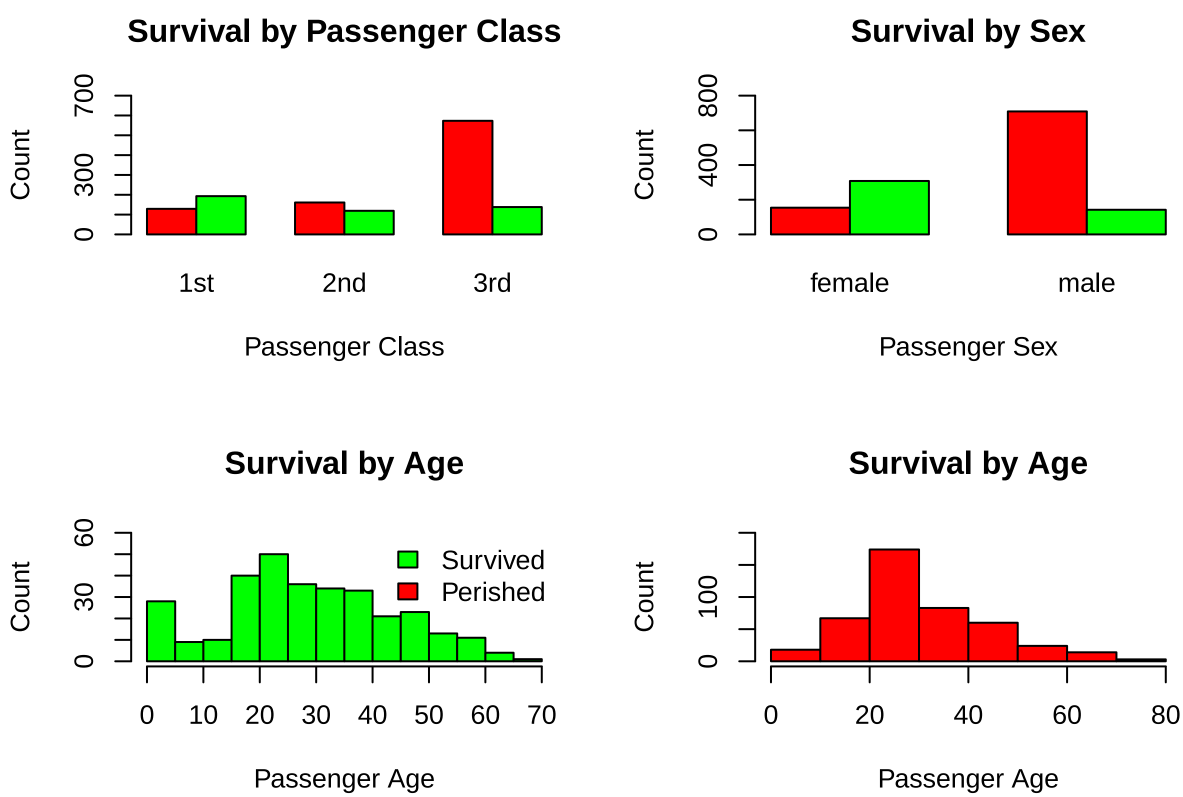
```
## Warning in plot.window(xlim, ylim, "", ...): "beside" bir grafiksel parametre
## değil
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
```

```
## "beside" bir grafiksel parametre değil

## Warning in axis(1, ...): "beside" bir grafiksel parametre değil

## Warning in axis(2, at = yt, ...): "beside" bir grafiksel parametre değil
```

```
legend("topright", legend=c("Survived", "Perished"), fill=c("green", "red"), pt.cex = 1.5, bty = "n")

hist(data$Age[data$Survived==0], beside = TRUE, main="Survival by Age",
        xlab="Passenger Age", ylab="Count", col="red", ylim=c(0,200))
```

```
## Warning in plot.window(xlim, ylim, "", ...): "beside" bir grafiksel parametre
## değil

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "beside" bir grafiksel parametre değil

## Warning in axis(1, ...): "beside" bir grafiksel parametre değil

## Warning in axis(2, at = yt, ...): "beside" bir grafiksel parametre değil
```



The plots tell us that most of those who perished were 3rd class passengers. Many more males died than females. Most of those who survived were around ages 20 and 30. Around the same ages most people perished too.

```
tot_survived <- xtabs(Survived ~ PClass + Sex, data=data)
tot_survived
```

```
##       Sex
## PClass female male
##    1st    134   59
```

```
##    2nd      94   25
##    3rd      80   58
```

```r
round(tot_survived/xtabs(~PClass+ Sex, data=data), 2)
```

```
##       Sex
## PClass female male
##    1st   0.94 0.33
##    2nd   0.88 0.14
##    3rd   0.38 0.12
```

```r
model <- glm(Survived ~ PClass + Age + Sex, data=data, family="binomial")
summary(model)$coefficients
```

```
##                Estimate  Std. Error    z value     Pr(>|z|)
## (Intercept)  3.75966210 0.397567324   9.456668 3.179129e-21
## PClass2nd   -1.29196240 0.260075781  -4.967638 6.777324e-07
## PClass3rd   -2.52141915 0.276656805  -9.113888 7.948131e-20
## Age         -0.03917681 0.007616218  -5.143868 2.691392e-07
## Sexmale     -2.63135683 0.201505379 -13.058494 5.684093e-39
```

Since all the probabilities of the variables are above zero, they are significant and can't be thrown out. The odds can be calculated using the estimate of this table like this: exp(3.76 + PClass2nd * -1.292 + PClass3rd * -2.521 + Age * -0.039 + Sexmale * -2.631) PClass and Sexmale are binary variables, while Age is continuous.

```r
# odds that a female 1st class passenger survived
exp(3.76)
```

```
## [1] 42.94843
```

```r
# odds that a 2nd class passenger survived
exp(3.76 + -1.292)
```

```
## [1] 11.79883
```

```r
# odds that someone who is 30yo survived
exp(3.76 + 30*-0.039)
```

```
## [1] 13.32977
```

```r
# odds that a male 3rd class passenger survived
exp(3.76 + -2.521 + -2.631)
```

```
## [1] 0.2485777
```

The above are some examples that can be calculated. The odds to survive for a first class female passenger are quite high, while it is low for a 3rd class male passenger.

Section b

```r
glm2 <- glm(Survived~Age*PClass, data=data, family=binomial)
anova(glm2, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
##
## Terms added sequentially (first to last)
```

```
##
##
##            Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                         755    1025.57
## Age         1    2.849       754    1022.72  0.09141 .
## PClass      2  112.807       752     909.92  < 2e-16 ***
## Age:PClass  2    1.166       750     908.75  0.55816
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
glm3 <- glm(Survived~Age*Sex, data=data, family=binomial)
anova(glm3, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                      755    1025.57
## Age      1    2.849       754    1022.72   0.09141 .
## Sex      1  227.138       753     795.59 < 2.2e-16 ***
## Age:Sex  1   25.030       752     770.56 5.645e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

H_0: All s are equal.

H_1: All s are not equal.

We study the interaction between Age and PClass. Only the last p-value is relevant, which is 0.56 and higher than significance level 0.05, therefore we do not reject null hypothesis, meaning that there is no interaction between Age and PClass. We do the same for Age and Sex: p-value is 5.645e-07, which is lower than 0.05, therefore we do reject null hypothesis, meaning that there is an interaction between Age and Sex.

We add this interaction to our model from Section a:

```r
model <- glm(Survived ~ PClass + Age + Sex + Age*Sex, data=data, family="binomial")
summary(model)$coefficients
```

```
##                Estimate Std. Error    z value     Pr(>|z|)
## (Intercept)  2.75656302 0.43764171  6.2986753 3.002001e-10
## PClass2nd   -1.54336652 0.28735776 -5.3708886 7.834962e-08
## PClass3rd   -2.65398052 0.29142296 -9.1069712 8.471384e-20
## Age          0.00244348 0.01140798  0.2141903 8.303986e-01
## Sexmale     -0.50818658 0.44251492 -1.1484055 2.508012e-01
## Age:Sexmale -0.07559126 0.01500877 -5.0364712 4.741923e-07
```

After adding interaction term to the original model, we get new p-values. This time, age and sex have p-values that are higher than our significance level, therefore these factors are no longer significant.

The new model is:

```r
model <- glm(Survived ~ PClass + Age:Sex, data=data, family="binomial")
summary(model)$coefficients
```

```
##                    Estimate  Std. Error   z value     Pr(>|z|)
## (Intercept)    2.50401090 0.377774141  6.628328 3.395105e-11
## PClass2nd     -1.58098832 0.288018409 -5.489192 4.037768e-08
## PClass3rd     -2.66612966 0.292351736 -9.119596 7.540503e-20
## Age:Sexfemale  0.01080822 0.008942977  1.208570 2.268280e-01
## Age:Sexmale   -0.08022407 0.009173972 -8.744747 2.235162e-18
```

And finally the probability of survival for each factor:

```r
age <- 55
pclass <- c("1st", "2nd", "3rd")
sex <- c("female", "male")

df <- expand.grid(PClass=pclass, Sex=sex, Age=age)

results = round(predict(model, newdata=df, type="response"), 3)

cbind(df, Survival_Prob=results)
```

```
##   PClass    Sex Age Survival_Prob
## 1    1st female  55         0.957
## 2    2nd female  55         0.820
## 3    3rd female  55         0.606
## 4    1st   male  55         0.129
## 5    2nd   male  55         0.030
## 6    3rd   male  55         0.010
```

Section c

To predict survival status and measure the quality of the prediction we could use a subset of the data as training data and another subset as a testing data. We could train the model using training data with glm(). Once the model is trained, we could make predictions with predict(). To measure the quality of the predictions we could use part of the testing data (without survival status) to predict and check how many matches we get. Then we divide the matches by a total number of passengers in the testing data, and we get a proportion of correct predictions. The closer this number is to 1 the higher the quality of the prediction.

Section d

We use 2-test to test for passenger class effect on the survival status, and Fisher's exact test to test for sex effect on the survival status, since Fisher's test is more suitable for 2x2 tables.

H_0: Passenger class has no effect on survival status.

```r
cont_table1 <- table(data$Survived, data$PClass)
cont_table2 <- table(data$Survived, data$Sex)

chisq.test(cont_table1)
```

```
##
##  Pearson's Chi-squared test
##
## data:  cont_table1
## X-squared = 172.3, df = 2, p-value < 2.2e-16
```

```r
fisher.test(cont_table2)
```

```
##
##  Fisher's Exact Test for Count Data
```

```
##
## data:  cont_table2
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.07620521 0.13155709
## sample estimates:
## odds ratio
##  0.1003494
```

Both tests indicate that both passenger class and sex have a significant effect on survival status.

Section e

Both approaches are used to test for different things. A contingency table is good for determining whether there is a relationship between two variables, like testing whether two factors are independent.

- **Advantages of 2-test:** easy to use, non-parametric (no assumption about distribution)
- **Disadvantages of 2-test:** needs a larger sample size, 80% of expected cell counts should be above 5, categorical data only.
- **Advantages of Fisher's Exact Test:** sample size can be relatively small, non-parametric (no assumption about distribution), robust against the violations of assumptions.
- **Disadvantages of Fisher's Exact Test:** 2x2 table only, is is less likely to reject null hypothesis compared to 2-test.

Logistic regression is used to model the relationship between response variable and predictor variable and can be used to predict the probability of a certain outcome.

- **Advantages of logistic regression:** wide range of predictor variables, such as continuous, categorical, ordinal, robust against outliers, odds ratios easy to interpret.
- **Disadvantages of logistic regression:** assumes linear relationship between predictor and outcome variables.

Therefore, no, the approach in d) is not wrong, it is just testing for different things.

Exercise 4: Military coups

Section a

H_0: Any subset of the s is equal to 0.

```r
coups <- read.table("coups.txt", header = TRUE)
coups$pollib <- as.factor(coups$pollib) # transform into factor
fit <- glm(miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
             numelec + numregim, data = coups, family = "poisson")
summary(fit)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn + size + numelec + numregim, family = "poisson", data = coups)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5075  -0.9533  -0.3100   0.4859   1.6459
##
## Coefficients:
```

```
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.2334274  0.9976112  -0.234  0.81500
## oligarchy    0.0725658  0.0353457   2.053  0.04007 *
## pollib1     -1.1032439  0.6558114  -1.682  0.09252 .
## pollib2     -1.6903057  0.6766503  -2.498  0.01249 *
## parties      0.0312212  0.0111663   2.796  0.00517 **
## pctvote      0.0154413  0.0101027   1.528  0.12641
## popn         0.0109586  0.0071490   1.533  0.12531
## size        -0.0002651  0.0002690  -0.985  0.32444
## numelec     -0.0296185  0.0696248  -0.425  0.67054
## numregim     0.2109432  0.2339330   0.902  0.36720
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 28.249  on 26  degrees of freedom
## AIC: 113.06
##
## Number of Fisher Scoring iterations: 5
```

From the output, we can see that oligarchy, political liberalization (pollib) and parties have a significant effect on the number of military coups, while other variables, such as size of the country and total number of legislative and presidential elections (numelec) do not have a significant effect.

Section b

In this section, we were asked to use the step-down approach to reduce the number of explanatory variables. To do that, we start by inspecting the summary of the full model (all explanatory variables included):

```
summary(glm(miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
              numelec + numregim, family=poisson, data=coups))
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn + size + numelec + numregim, family = poisson, data = coups)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5075  -0.9533  -0.3100   0.4859   1.6459
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.2334274  0.9976112  -0.234  0.81500
## oligarchy    0.0725658  0.0353457   2.053  0.04007 *
## pollib1     -1.1032439  0.6558114  -1.682  0.09252 .
## pollib2     -1.6903057  0.6766503  -2.498  0.01249 *
## parties      0.0312212  0.0111663   2.796  0.00517 **
## pctvote      0.0154413  0.0101027   1.528  0.12641
## popn         0.0109586  0.0071490   1.533  0.12531
## size        -0.0002651  0.0002690  -0.985  0.32444
## numelec     -0.0296185  0.0696248  -0.425  0.67054
## numregim     0.2109432  0.2339330   0.902  0.36720
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 28.249  on 26  degrees of freedom
## AIC: 113.06
##
## Number of Fisher Scoring iterations: 5
```

We see that variable *numelec* has the biggest p-value and not significant (< 0.05). Hence we remove it from the model and proceed with the next step of the method:

```
summary(glm(miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
              numregim, family=poisson, data=coups))
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn + size + numregim, family = poisson, data = coups)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5346  -0.9405  -0.3131   0.4241   1.6642
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.4577458  0.8602345  -0.532  0.59464
## oligarchy    0.0812015  0.0288154   2.818  0.00483 **
## pollib1     -0.9642976  0.5620939  -1.716  0.08625 .
## pollib2     -1.5149509  0.5269441  -2.875  0.00404 **
## parties      0.0293409  0.0103101   2.846  0.00443 **
## pctvote      0.0139115  0.0094654   1.470  0.14164
## popn         0.0099592  0.0067249   1.481  0.13862
## size        -0.0002688  0.0002687  -1.000  0.31710
## numregim     0.1804415  0.2241166   0.805  0.42075
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 28.430  on 27  degrees of freedom
## AIC: 111.24
##
## Number of Fisher Scoring iterations: 5
```

We see that variable *numregim* has the biggest p-value and not significant (< 0.05). Hence we remove it from the model and proceed with the next step of the method:

```
summary(glm(miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size,
            family=poisson, data=coups))
```

```
##
## Call:
```

```
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn + size, family = poisson, data = coups)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5513  -0.8958  -0.2225   0.5258   1.6058
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.0419757  0.5774100   0.073 0.942048
## oligarchy    0.0894951  0.0270440   3.309 0.000936 ***
## pollib1     -0.9673253  0.5605601  -1.726 0.084412 .
## pollib2     -1.5321126  0.5232779  -2.928 0.003412 **
## parties      0.0288170  0.0102173   2.820 0.004796 **
## pctvote      0.0149216  0.0093762   1.591 0.111513
## popn         0.0071647  0.0056842   1.260 0.207510
## size        -0.0002579  0.0002662  -0.969 0.332621
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 29.081  on 28  degrees of freedom
## AIC: 109.89
##
## Number of Fisher Scoring iterations: 5
```

We see that variable *size* has the biggest p-value and not significant (< 0.05). Hence we remove it from the model and proceed with the next step of the method:

```
summary(glm(miltcoup ~ oligarchy + pollib + parties + pctvote + popn,
            family=poisson, data=coups))
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn, family = poisson, data = coups)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4197  -0.9952  -0.1443   0.5699   1.6107
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.231435   0.528887  -0.438  0.66168
## oligarchy    0.083468   0.025829   3.232  0.00123 **
## pollib1     -0.683589   0.495822  -1.379  0.16799
## pollib2     -1.320568   0.490268  -2.694  0.00707 **
## parties      0.029770   0.010310   2.887  0.00388 **
## pctvote      0.013925   0.009371   1.486  0.13728
## popn         0.005659   0.005483   1.032  0.30204
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 30.040  on 29  degrees of freedom
## AIC: 108.85
##
## Number of Fisher Scoring iterations: 5
```

We see that variable *popn* has the biggest p-value and not significant (< 0.05). Hence we remove it from the model and proceed with the next step of the method:

```r
summary(glm(miltcoup ~ oligarchy + pollib + parties + pctvote,
            family=poisson,data=coups))
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote,
##     family = poisson, data = coups)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5300  -0.9794  -0.1833   0.5662   1.6721
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.116499   0.513751  -0.227  0.82061
## oligarchy    0.094712   0.023184   4.085  4.4e-05 ***
## pollib1     -0.620756   0.487526  -1.273  0.20292
## pollib2     -1.310374   0.489017  -2.680  0.00737 **
## parties      0.025745   0.009552   2.695  0.00704 **
## pctvote      0.012057   0.009072   1.329  0.18383
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 31.069  on 30  degrees of freedom
## AIC: 107.88
##
## Number of Fisher Scoring iterations: 5
```

We see that variable *pctvote* has the biggest p-value and not significant (< 0.05). Hence we remove it from the model and proceed with the next step of the method:

```r
summary(glm(miltcoup ~ oligarchy + pollib + parties, family=poisson, data=coups)) # final model
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties, family = poisson,
##     data = coups)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3609  -1.0407  -0.3153   0.6145   1.7536
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.207981   0.445679   0.467   0.6407
## oligarchy    0.091466   0.022563   4.054 5.04e-05 ***
## pollib1     -0.495414   0.475645  -1.042   0.2976
## pollib2     -1.112086   0.459492  -2.420   0.0155 *
## parties      0.022358   0.009098   2.458   0.0140 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 32.822  on 31  degrees of freedom
## AIC: 107.63
##
## Number of Fisher Scoring iterations: 5
```

We see that all remaining variables are significant so we stop the procedure. Consequently, the resulting model would include the following explanatory variables: *oligarchy*, *pollib* and *parties* Note that this is the exact same result that we obtained in Section a, where we deemed these 3 variables as significant.

Section c

```r
is.factor(coups$pollib) # double-check it is a factor
```

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

```r
mean(coups$parties); mean(coups$oligarchy) # which gives 5.22 and 17.08
```

```
## [1] 17.08333
```

```
## [1] 5.222222
```

```r
coupnew <- data.frame(pollib=c("0", "1", "2"), oligarchy=c(5.22, 5.22, 5.22),
                      parties=c(17.08, 17.08, 17.08))
modelf <- glm(miltcoup ~ oligarchy + pollib + parties, family=poisson, data=coups)
predict(modelf, coupnew, type = "response")
```

```
##        1        2        3
## 2.907544 1.771620 0.956210
```

Our model predicts that the amount of expected coups decreases as the level of political liberalization increases, with pollib = 0 having 2.91 successful coups, pollib = 1 having 1.77 successful coups, and pollib = 2 having less than 1 political coup.