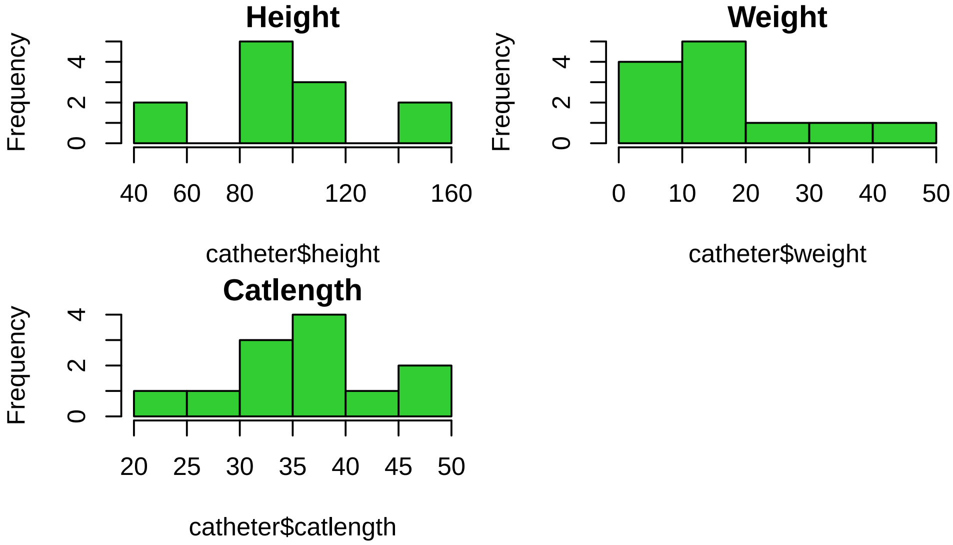
Solution to Series 4

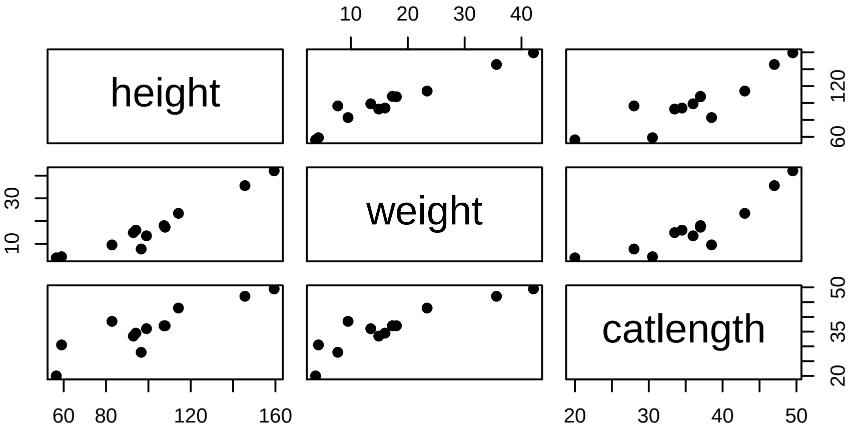
1. a)

```
> ## load data
> load("catheter.rda")
> ## histograms
> par(mfrow=c(2,2))
> hist(catheter$height, col="limegreen", main="Height")
> hist(catheter$weight, col="limegreen", main="Weight")
> hist(catheter$catlength, col="limegreen", main="Catlength")
```



First of all, note that only 12 data points are available. This is the lower limit for a multiple linear regression with two predictors as there are only four observations per parameter. Secondly, all variables take positive values only so that log transformations would be feasible. However, we only see a skewed distribution for the variable `weight`. This time we will not do a transformation even though it could be benefitial.

```
> ## pairs plot
> pairs(catheter, pch=19)
```



In the pairs plot we see that `height` and `weight` correlate strongly. This is not surprising as the observations come from children. Similarly, there is a strong relation between the target `catlength` and the two predictors.

b) The predictor is highly significant in both cases:

```
> ## simple linear regressions
> fits1 <- lm(catlength ~ height, data=catheter)
> fits2 <- lm(catlength ~ weight, data=catheter)
> summary(fits1)

Call:
lm(formula = catlength ~ height, data = catheter)

Residuals:
    Min      1Q  Median      3Q     Max
-7.0929 -0.7298 -0.2608  1.1652  6.6879

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.12706    4.24700   2.855 0.017090 *
height       0.23774    0.04034   5.893 0.000152 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.009 on 10 degrees of freedom
Multiple R-squared:  0.7764,       Adjusted R-squared:  0.7541
F-statistic: 34.73 on 1 and 10 DF,  p-value: 0.0001525

> summary(fits2)

Call:
lm(formula = catlength ~ weight, data = catheter)

Residuals:
    Min      1Q  Median      3Q     Max
-7.9676 -1.4963 -0.1386  2.0980  7.0205

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 25.62631    2.00264  12.796 1.59e-07 ***
weight       0.61613    0.09759   6.313 8.75e-05 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.797 on 10 degrees of freedom
Multiple R-squared:  0.7994,       Adjusted R-squared:  0.7794
F-statistic: 39.86 on 1 and 10 DF,  p-value: 8.755e-05
```

c)

```
> ## multiple regression
> fit   <- lm(catlength ~ height + weight, data=catheter)
> summary(fit)

Call:
lm(formula = catlength ~ height + weight, data = catheter)

Residuals:
    Min      1Q  Median      3Q     Max
-7.0497 -1.2753 -0.2595  1.9095  6.9933

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 21.08527    8.77037   2.404   0.0396 *
height       0.07681    0.14412   0.533   0.6070
weight       0.42752    0.36810   1.161   0.2753
```

```
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.94 on 9 degrees of freedom
Multiple R-squared:  0.8056,	Adjusted R-squared:  0.7624
F-statistic: 18.65 on 2 and 9 DF,  p-value: 0.0006301
```

Yes, there is an influence of the predictors on the target variable overall. This is assessed by the global F-test. Its p-value is smaller than 0.01 so that the null hypothesis is rejected at the 1% level. At least one of the predictors is necessary.

d) As we can see from the summary output (see above), both null hypotheses are retained, i.e. the predictors are not significant. Is this a contradiction to the results from the two simple linear regressions? No – in multiple regression the hypotheses tests assess whether we need (e.g.) the predictor `height` when we already know the predictor `weight`. The answer is no and the same holds vice versa. On the other hand, the global F-test indicates that we need at least one of the two predictors. So we do not need to include both predictors simultaneously but we need one of them. This situation occurs when the predictors are strongly correlated. Due to the smaller p-value we would prefer the predictor `weight` in this case.

e)

```
> ## prediction intervals
> newdat <- data.frame(height=120, weight=25)
> predict(fits1, newdata=newdat, interval="prediction")

       fit      lwr      upr
1 40.65609 31.20891 50.10327

> predict(fits2, newdata=newdat, interval="prediction")

       fit      lwr      upr
1 41.02954 32.06162 49.99747

> predict(fit,   newdata=newdat, interval="prediction")

       fit      lwr      upr
1 40.99072 31.53989 50.44154
```

The predictions differ slightly. We note that the prediction interval is not shortest for the multiple regression model which one might expect since it uses the largest amount of information. However, the multiple model requires estimating one additional parameter based on the available 12 data points. This is associated with a larger estimation error of each single parameter. In most practical cases the prediction accucacy increases by including an additional parameter but in our case the increased estimation error has a stronger, negative influence. This is due to the fact that the two predictors are strongly correlated – adding the second predictor when the first one is already present does hardly yield additional information.

In practice, a prediction error of ± 2cm would be acceptable. Thus, the data and the models do not allow for a prediction of `catlength` that is sufficiently precise.

2. a) The model shows a large systematic error, i.e. in the Tukey-Anscombe plot the smoother deviates massively from the x-axis. In addition, the distribution of the residuals is skewed.

b) For the salt content there is an optimal amount. If the cake contains too little salt the cake does not have a lot of taste which yields a smaller score. If the cake contains too much salt, the cake does not taste well either. Therefore, the parameter needs to be negative – if the salt content exceeds the optimum, the score decreases and if the salt content is smaller than the optimum the score decreases as well.

c) There are 46 degrees of freedom and four parameters are estimated (as there are three predictors). Thus, there are 50 observations.

d) The fitted value on the scale of the logarithmic score is:

```
> -0.4150 + 4.0609*3.5 + (-1.0725)*3.5^2 + 2.0109*1
[1] 2.670925
```

To compute the conditional median, we reverse the log transformation:

```
> exp(2.670925)
[1] 14.45333
```

To compute the conditional expectation, we need to add $\frac{1}{2}\sigma_E^2$, i.e.

```
> exp(2.670925 + 0.5*2.784^2)
[1] 696.629
```

e) The null hypothesis is $H_0 : \beta_4 = 0$ and we test against the alternative $H_A : \beta_4 \neq 0$. We use the test statistic for the partial F-test:

$$F = \frac{n-(p+1)}{p-q} \cdot \frac{RSS_{small} - RSS_{big}}{RSS_{big}} \sim F_{p-q,n-(p+1)}$$

To compute the observed value of the test statistic, we need the residual sum of squares. It can be computed from the residual standard error. For the small model we get: $46 \cdot 2.784^2 = 356.53$ and for the big model we get: $45 \cdot 2.7^2 = 328.05$. For the test statistic we then get a value of 3.907. The p-value is:

```
> 1-pf(3.907, 1, 45)
[1] 0.05423324
```

So the null hypothesis is retained as this value is (slightly) larger than 0.05. The term that was added in the new model is not significant and does not have to be included in the model.

3. **a)**

```
> ## load data
> load("conconi2.rda")
> ## preprocess
> speed  <- conconi2$Speed[c(1:19,7:26)]
> puls   <- c(conconi2$Marcel.Puls[1:19], conconi2$Dani.Puls[7:26])
> runner <- factor(c(rep("Marcel",19), rep("Dani",20)))
> c2     <- data.frame(puls, speed, runner)
```

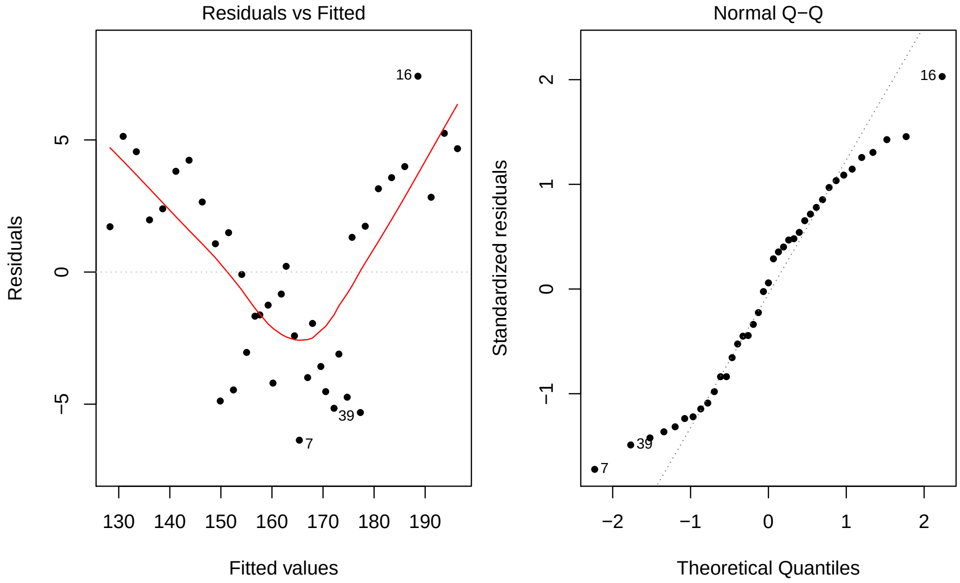
b)

```
> ## perform regression
> fit1  <- lm(puls ~ speed + runner, data=c2)
> summary(fit1)

Call:
lm(formula = puls ~ speed + runner, data = c2)

Residuals:
   Min     1Q Median     3Q    Max 
-6.364 -3.340  0.217  2.992  7.411 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   66.3510     3.7310   17.78   <2e-16 ***
speed          5.1611     0.2169   23.80   <2e-16 ***
runnerMarcel  37.0789     1.4096   26.30   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.811 on 36 degrees of freedom
Multiple R-squared:  0.959,	Adjusted R-squared:  0.9568 
F-statistic: 421.5 on 2 and 36 DF,  p-value: < 2.2e-16
```

The main effects model assumes that both runners are identical w.r.t. the increase in pulse while the initial pulse can differ.

c)

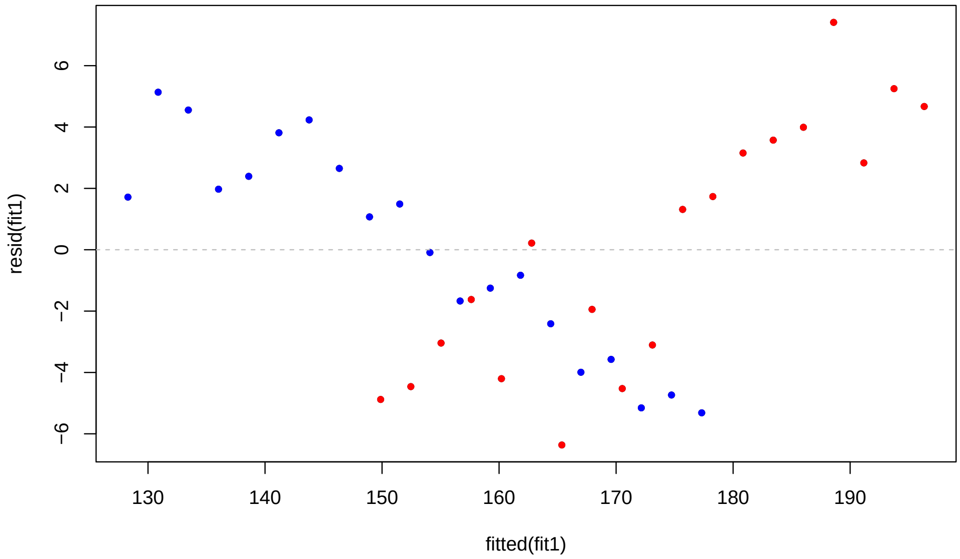
```
> ## residual analysis
> par(mfrow=c(1,2))
> plot(fit1, which=1:2, pch=20)
```



We observe a large systematic error. In addition, the distribution of the residuals is short-tailed in comparison to a Normal distribution. We need to fix the systematic error which can often be achieved by a variable transformation. In this case, however, it is more plausible that the error is rooted in a model mispecification and can be fixed by including the interaction term.

d)

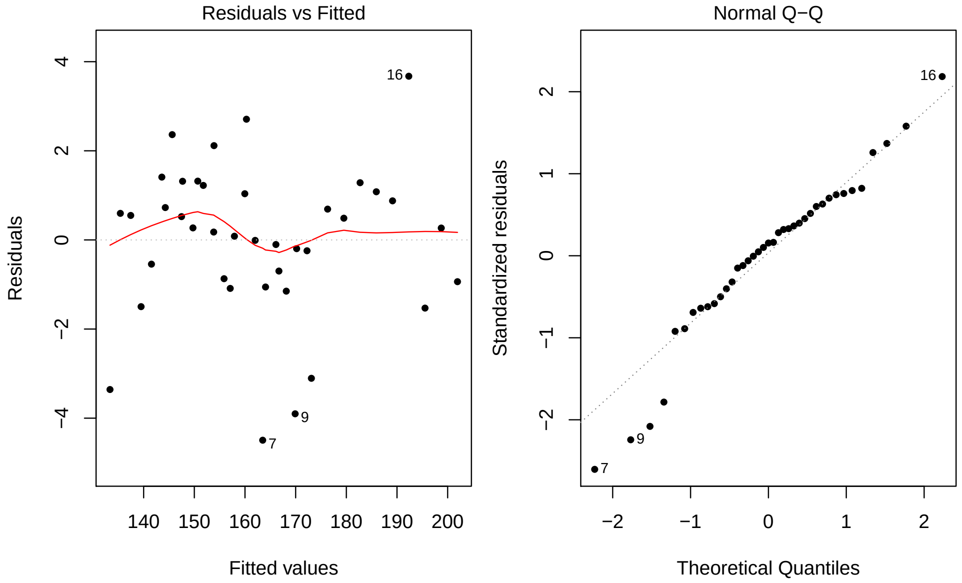
```
> ## residual analysis
> par(mfrow=c(1,1))
> plot(fitted(fit1), resid(fit1), pch=20)
> weli <- which(c2$runner=="Marcel")
> points(fitted(fit1)[weli], resid(fit1)[weli], pch=20, col="red")
> weli1 <- which(c2$runner=="Dani")
> points(fitted(fit1)[weli1], resid(fit1)[weli1], pch=20, col="blue")
> abline(h=0, col="grey", lty=2)
```



In case of Dani's run (blue points), the pulse is underestimated at small values while for Marcel's run it is vice versa. This indicates that we cannot use two parallel regression lines but need to consider a more complex model with an interaction term. This allows for different slopes of the regression lines.

e) This model assumes different initial pulses as well as different slopes, i.e. two distinct regression lines are fitted.

```
> ## new model
> fit2 <- lm(puls ~ speed + runner + speed:runner, data=c2)
> ## residual analysis
> par(mfrow=c(1,2))
> plot(fit2, which=1:2, pch=20)
```



The model fits well. There is only a small deviation of the smoother from the x-axis which can be tolerated. There are four outliers in the Normal plot, i.e. four observations with large negative residuals. These deviations were already discussed previously: recall that they were caused by not matching the required speed exactly.

f)
```
> summary(fit2)
```

```
Call:
lm(formula = puls ~ speed + runner + speed:runner, data = c2)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.4947 -0.9034  0.2667  1.0588  3.6737 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)         84.2383     2.3574  35.734  < 2e-16 ***
speed                4.0932     0.1387  29.512  < 2e-16 ***
runnerMarcel         2.3722     3.1330   0.757    0.454    
speed:runnerMarcel   2.3138     0.2042  11.333 2.91e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.788 on 35 degrees of freedom
Multiple R-squared:  0.9912,	Adjusted R-squared:  0.9905 
F-statistic:  1319 on 3 and 35 DF,  p-value: < 2.2e-16
```

The initial pulse of Dani corresponds to the intercept, i.e. 84.2. For Marcel, the coefficient $\hat{\beta}_2 = 2.4$ needs to be added, so his initial pulse is 86.6. For Dani, the pulse increases by 4.1 beats with every additional km/h in speed. For Marcel, the coefficient $\hat{\beta}_3 = 2.3$ needs to be added so that we get a value of 6.4. The p-value for $H_0 : \beta_3 = 0$ is $3 \cdot 10^{-13}$, i.e. the difference is highly significant.

4. a) First, we check the structure of the data frame:

```
> ## load data
> load("farm.rda")
> ## check properties of the data
> str(farm)
```

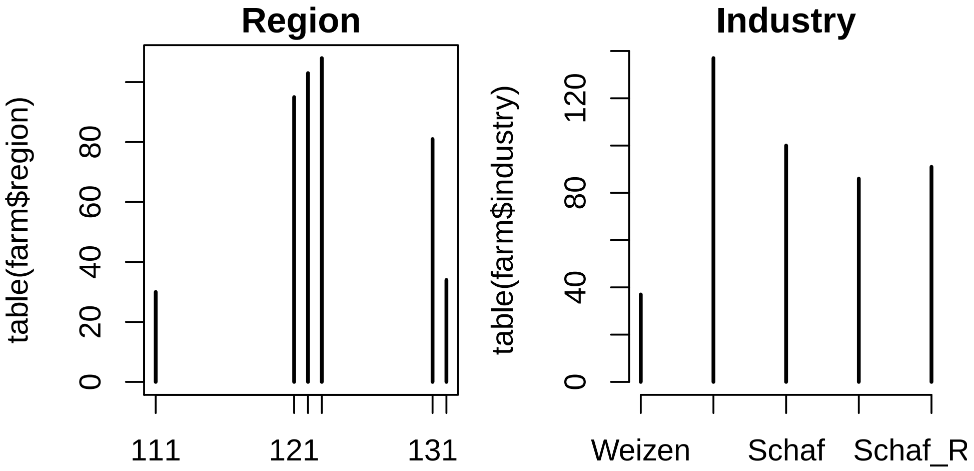
```
'data.frame':	451 obs. of  4 variables:
 $ region  : int  111 111 111 111 111 111 111 111 111 111 ...
 $ industry: int  3 5 2 1 2 5 2 3 3 3 ...
 $ aufwand : int  115096 75443 378857 433590 347417 327745 714462 221258 241868 194837 ...
 $ ertrag  : int  147652 82920 442726 649628 407836 472569 576372 241864 339215 356625 ...
```

All variables are of data type “int”. This is incorrect for the factor variables `region` and `industry` and would lead to incorrect regression results. We define the factor variables as follows:

```
> farm$region   <- factor(farm$region)
> farm$industry <- factor(farm$industry, labels=c("Weizen", "Weizen_Schaf_Rind", "Schaf", "Rin
```

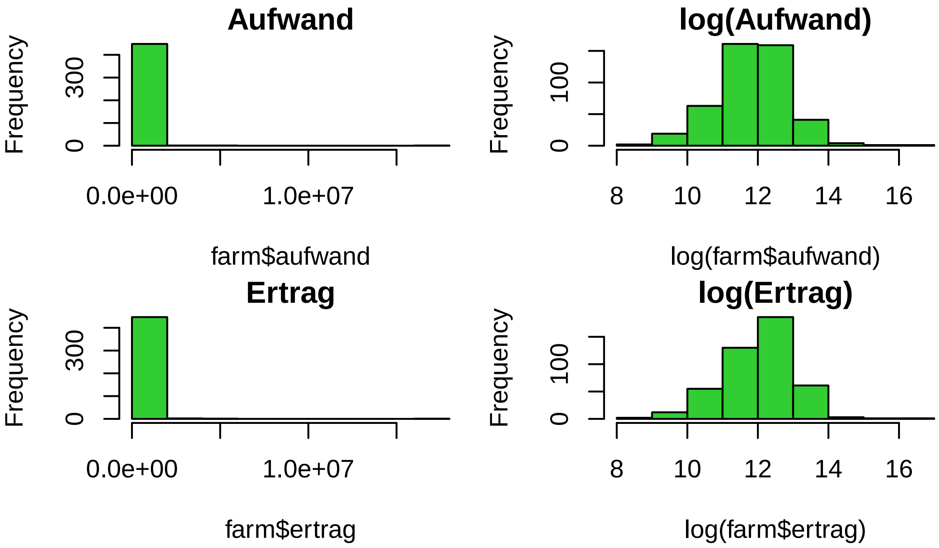
We now check whether transformations are needed and whether there are sufficiently many observations for all levels of the factor variables. The recommendation is that there are at least five observations for each level.

```
> ## visualization
> par(mfrow=c(1,2))
> plot(table(farm$region), main="Region")
> plot(table(farm$industry), main="Industry")
```



The number of observations are sufficient for all levels of the factor variables.

```
> ## visualization
> par(mfrow=c(2,2))
> hist(farm$aufwand, main="Aufwand", col="limegreen")
> hist(log(farm$aufwand), main="log(Aufwand)", col="limegreen")
> hist(farm$ertrag, main="Ertrag", col="limegreen")
> hist(log(farm$ertrag), main="log(Ertrag)", col="limegreen")
```



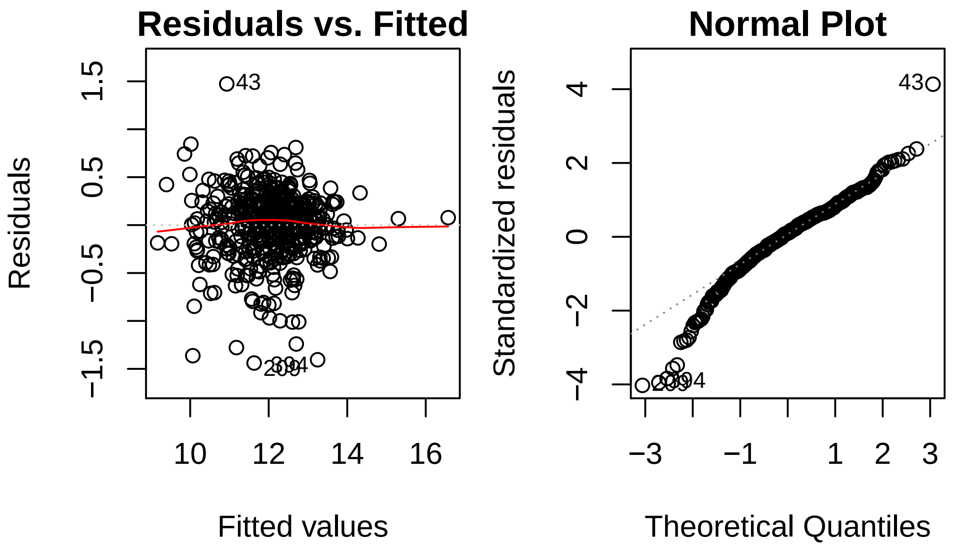
The plots show that we need to apply a log transformations.

b)

```
> ## fit main effects model
> fit <- lm(log(ertrag) ~ log(aufwand) + region + industry, data=farm)
> summary(fit)
```

```
Call:
lm(formula = log(ertrag) ~ log(aufwand) + region + industry,
    data = farm)

Residuals:
     Min       1Q   Median       3Q      Max
-1.43881 -0.17143  0.03773  0.22168  1.47317

Coefficients:
                           Estimate Std. Error t value
(Intercept)                1.379636   0.248432   5.553
log(aufwand)               0.917954   0.018617  49.306
region121                 -0.076883   0.077353  -0.994
region122                 -0.082997   0.076912  -1.079
region123                 -0.036680   0.076151  -0.482
region131                 -0.003855   0.079775  -0.048
region132                 -0.243938   0.100536  -2.426
industryWeizen_Schaf_Rind -0.155614   0.068023  -2.288
industrySchaf             -0.222879   0.071421  -3.121
industryRind               0.002649   0.075844   0.035
industrySchaf_Rind        -0.171106   0.072947  -2.346
                          Pr(>|t|)
(Intercept)               4.86e-08 ***
log(aufwand)               < 2e-16 ***
region121                  0.32081
region122                  0.28113
region123                  0.63027
region131                  0.96148
region132                  0.01565 *
industryWeizen_Schaf_Rind  0.02263 *
industrySchaf              0.00192 **
industryRind               0.97215
industrySchaf_Rind         0.01944 *
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3612 on 440 degrees of freedom
Multiple R-squared:  0.8712,        Adjusted R-squared:  0.8683
F-statistic: 297.7 on 10 and 440 DF,  p-value: < 2.2e-16
```

```
> ## residual analysis
> par(mfrow=c(1,2))
> plot(fit, which=1, caption="", main="Residuals vs. Fitted")
> plot(fit, which=2, caption="", main="Normal Plot")
```



The Tukey-Anscombe plot does not indicate the presence of a systematic error. The Normal plot shows that the distribution of the residuals is skewed to the left and there is one large positive outlier (no. 43). In summary, the assumptions seem to be fulfilled to a sufficient degree but not entirely.

c)

```
> ## predict
> newdat <- data.frame(aufwand=10^5, region="111", industry="Rind")
> predi  <- predict(fit, newdata=newdat)
> exp(predi + 0.5*summary(fit)$sigma^2)

       1
165357.7
```

Using `predict()` we obtain the prediction on the log scale. We thus need to transform the value back to the original scale. So the expected revenue is 165'357.7 Dollar.

d)

```
> drop1(fit, test="F")

Single term deletions

Model:
log(ertrag) ~ log(aufwand) + region + industry
             Df Sum of Sq    RSS     AIC   F value
<none>                     57.41 -907.62
log(aufwand)  1    317.21 374.62  -63.69 2431.0923
region        5      1.36  58.77 -907.03    2.0906
industry      4      2.77  60.18 -894.39    5.3007
                Pr(>F)
<none>
log(aufwand) < 2.2e-16 ***
region       0.0655074 .
industry     0.0003542 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The predictor `region` is not significant as can be seen from the p-value 0.0655 of the partial F-test.

e)

- 31 parameters are estimated as the model has 420 degrees of freedom and there are 451 observations.
- We have sufficiently many observations as there are more than five observations for every estimated parameter.
- To test the interaction term we need to do a partial F-test. We could do this explicitly with the command `anova()` but using `drop1()` is more convenient:

```
> ## option 1
> f.big   <- lm(log(ertrag) ~ log(aufwand) + region + industry + region:industry, data=farm)
> f.small <- lm(log(ertrag) ~ log(aufwand) + region + industry, data=farm)
> anova(f.small, f.big)
```

```
Analysis of Variance Table

Model 1: log(ertrag) ~ log(aufwand) + region + industry
Model 2: log(ertrag) ~ log(aufwand) + region + industry + region:industry
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    440 57.411
2    420 54.540 20    2.8706 1.1053 0.3404
> # option 2
> drop1(f.big, test="F")
Single term deletions

Model:
log(ertrag) ~ log(aufwand) + region + industry + region:industry
                Df Sum of Sq    RSS     AIC   F value
<none>                        54.54 -890.75
log(aufwand)     1   303.467 358.01  -44.14 2336.9109
region:industry 20     2.871  57.41 -907.62    1.1053
                Pr(>F)
<none>
log(aufwand)    <2e-16 ***
region:industry 0.3404
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The interaction term is not significant and can be excluded from the model.

- The intuitive meaning of the interaction term is that region and industry do not influence revenue independently and additively but that the influence of industry differs between regions. However, as we have seen this is not the case for this data set.

f) The interaction term is not significant as we have seen above. So we exclude it and are left with the main effects model. Also for this model, we have seen that `region` is not significant, so we will exclude it as well. This leaves the model where the (logarithmic) revenue is explained with the (logarithmic) costs and the industry. In this model both predictors are significant. That is why we decide to use this model.