Solution to exercise about process loss in course 02935

Elisabeth Wreford Andersen

12 June 2018

1 Introduction

The dataset process.txt contains measurements of air flow, water temperature, and acid concentration of a process loss. The aim of this case is to explain the process loss as a function of the other variables.

Data:

Variable	Description
loss	loss from process
airflow	air flow watertemp water temperature
acidconc	acid concentration

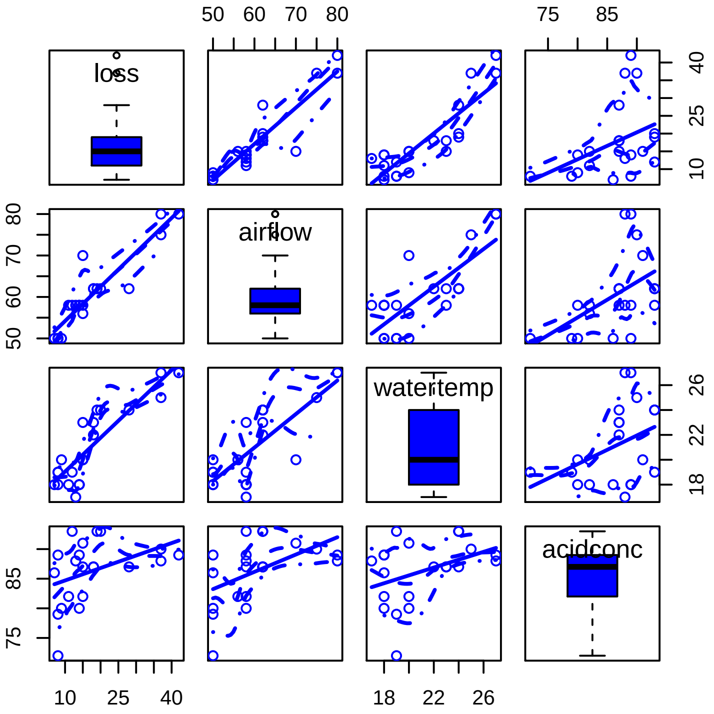
Exercise

1. Plot the variables and make a graphical assessment. Which variables could be helpful in explaining process loss?

First we will read in the data and make a scatterplot of all the variables.

```
setwd("C:/ewan/Teaching/PhD_applied_statistics_J18/MultipleRegression/Data")


process <- read.delim("process.txt")
str(process)
```

```
## 'data.frame': 21 obs. of  4 variables:
##  $ loss     : int  42 37 37 28 18 18 19 20 15 14 ...
##  $ airflow  : int  80 80 75 62 62 62 62 62 58 58 ...
##  $ watertemp: int  27 27 25 24 22 23 24 24 23 18 ...
##  $ acidconc : int  89 88 90 87 87 87 93 93 87 80 ...

library(car)
scatterplotMatrix(process,  diagonal=list(method="boxplot"))
```



Figur 1: Scatterplot for process data

From Figure 1 we see that loss seems to be related to air flow, water temperature and to some degree also acid concentration. Especially air flow and water temperature seem (highly) correlated.

2. Using simple linear regression, assess whether air flow, water temperature and acid concentration have an influence on process loss.

To do this we will fit three separate linear regressions:

```
fm.air <- lm(loss ~ airflow, data = process)
fm.temp <- lm(loss ~ watertemp, data = process)
fm.acid <- lm(loss ~ acidconc, data = process)
summary(fm.air)

##
## Call:
## lm(formula = loss ~ airflow, data = process)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.290  -1.127  -0.046   1.117   8.873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -44.13       6.11   -7.23  7.3e-07 ***
## airflow         1.02       0.10   10.21  3.8e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.1 on 19 degrees of freedom
## Multiple R-squared:  0.846,Adjusted R-squared:  0.838
## F-statistic:  104 on 1 and 19 DF,  p-value: 3.77e-09

summary(fm.temp)

##
## Call:
## lm(formula = loss ~ watertemp, data = process)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.890 -3.621  0.379  2.840  8.475
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -41.911      7.606   -5.51  2.6e-05 ***
## watertemp      2.817      0.357    7.90  2.0e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.04 on 19 degrees of freedom
## Multiple R-squared:  0.767,Adjusted R-squared:  0.754
## F-statistic: 62.4 on 1 and 19 DF,  p-value: 2.03e-07

summary(fm.acid)

##
## Call:
## lm(formula = loss ~ acidconc, data = process)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11.58  -5.58  -3.07   1.25  22.42
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -47.963     34.504   -1.39    0.181
## acidconc       0.759      0.399    1.90    0.073 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.57 on 19 degrees of freedom
## Multiple R-squared:  0.16,Adjusted R-squared:  0.116
## F-statistic: 3.62 on 1 and 19 DF,  p-value: 0.0725
```

From these analyses we see that both airflow and watertemp are highly significant but not acidconc.

3. Now use a multiple linear regression to assess the effects of air flow, water temperature and acid concentration on process loss. Notice what happens to the significance of the variables: One of the variables was (borderline) significant in the simple linear regression, but is not significant in the multiple linear regression. How do you explain this?

Instead of entering the variables one at a time we will try to do a multiple regression where all three variables are included as explanatory variables.

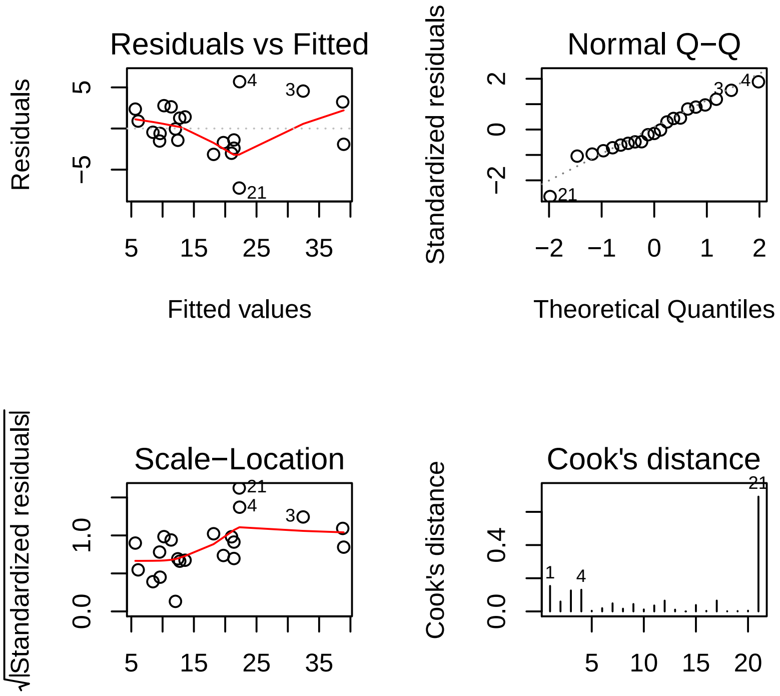
```
fm.all <- lm(loss ~ airflow + watertemp + acidconc, data = process)
summary(fm.all)

##
## Call:
```

```
## lm(formula = loss ~ airflow + watertemp + acidconc, data = process)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.238 -1.712 -0.455  2.361  5.698
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -39.920     11.896   -3.36   0.0038 **
## airflow        0.716      0.135    5.31  5.8e-05 ***
## watertemp      1.295      0.368    3.52   0.0026 **
## acidconc      -0.152      0.156   -0.97   0.3440
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.24 on 17 degrees of freedom
## Multiple R-squared:  0.914,Adjusted R-squared:  0.898
## F-statistic: 59.9 on 3 and 17 DF,  p-value: 3.02e-09

drop1(fm.all, test = "F")

## Single term deletions
##
## Model:
## loss ~ airflow + watertemp + acidconc
##           Df Sum of Sq RSS  AIC F value  Pr(>F)
## <none>                 179 53.0
## airflow    1       296 475 71.5   28.16 5.8e-05 ***
## watertemp  1       130 309 62.5   12.39  0.0026 **
## acidconc   1        10 189 52.1    0.95  0.3440
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the above analysis we see that acidconc is not at all significant for the loss ($p = 0.3440$). In the simple model acidconc was borderline significant with $p = 0.073$. The multiple regression model now takes all three variables into account and it seems that when we know airflow and watertemp then acidconc is no longer useful. From the scatter plot in Figure 1 we also see that acidconc is related to both airflow and watertemp, so the borderline effect we see in the simple analysis was due to this connection.

However, before we interpret our model too much we must check the underlying assumptions (independence, variance homogeneity and normal residuals). We will have to assume that the data sample was chosen so the observations can be assumed independent the other two assumptions are checked in Figure 2.

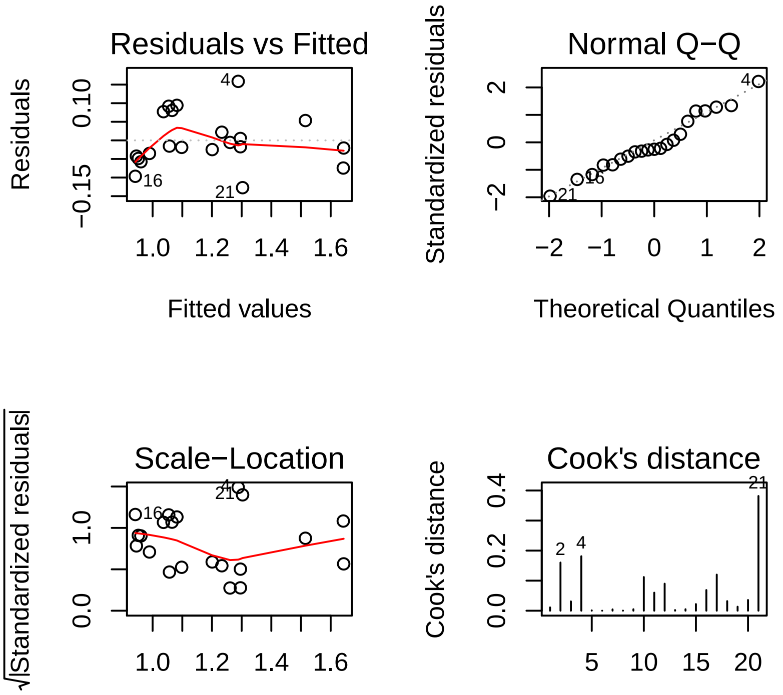
```
par(mfrow=c(2, 2))
plot(fm.all, which=1:4)
par(mfrow=c(1, 1))
```



Figur 2: Plots for checking initial model

From the plot in the top left of Figure 2 we see a trumpet shape and also the plot in the bottom left shows a line that is not horizontal. This all indicates a problem with variance homogeneity and I will log-transform the outcome:

```
fm3 <- lm(log10(loss) ~ airflow + watertemp + acidconc, data = process)
par(mfrow=c(2, 2))
plot(fm3, which=1:4)
par(mfrow=c(1, 1))
```



Figur 3: Plots for checking log-transformed model

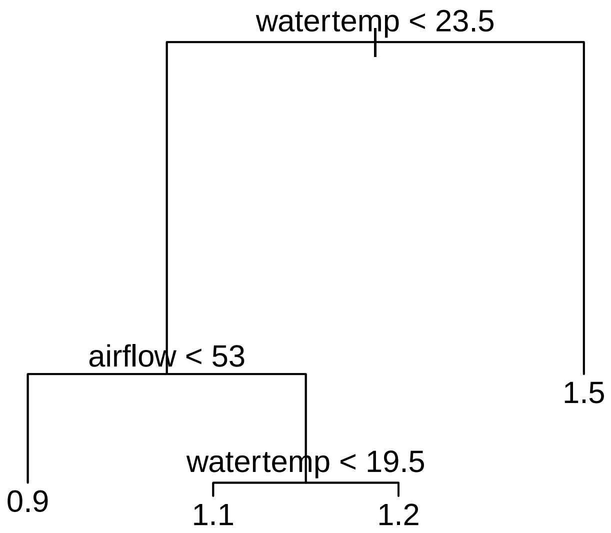
From Figure 3 we see that the trumpet shape has been much reduced and the assumption about normal residuals is also ok (the plot in the top right looks like a straight line).

Perhaps this model is too simple, so I will add all the squared terms to the model. We can also add the three possible interactions. However, we only have 21 observations, so we must be careful not to add too many explanatory variables at a time.

We can use a tree to get an idea about possible interactions, unfortunately, we do not have enough data to use the GAM to get ideas about the curves.

```
library(tree)
model<-tree(log10(loss) ~., data = process)
plot(model)
```

```
text(model)
```



Figur 4: Tree for log-transformed data

From Figure 4 there might be an interaction between watertemp and airflow. We see higher values of log-loss for higher values of watertemp. I will also add the squared terms and then try to simplify the model by backwards selection:

```
fm4 <- lm(log10(loss) ~ airflow + watertemp + acidconc + I(airflow^2) +
            I(watertemp^2) + I(acidconc^2) + airflow:watertemp,
          data = process)
drop1(fm4, test = "F")

## Single term deletions
##
## Model:
## log10(loss) ~ airflow + watertemp + acidconc + I(airflow^2) +
##     I(watertemp^2) + I(acidconc^2) + airflow:watertemp
##                    Df Sum of Sq    RSS  AIC F value Pr(>F)
## <none>                          0.0446 -113
## acidconc            1   0.00220 0.0468 -114    0.64  0.438
## I(airflow^2)        1   0.02313 0.0677 -106    6.75  0.022 *
## I(watertemp^2)      1   0.00001 0.0446 -115    0.00  0.951
## I(acidconc^2)       1   0.00255 0.0471 -114    0.74  0.404
## airflow:watertemp   1   0.00488 0.0495 -113    1.42  0.254
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From this analysis we decide to remove $I(watertemp^2)$ as the p-value is 0.951 so squared watertemp is not statistically significant. We will continue in this way, removing one variable at a time until all variables are statistically significant.

```
fm5 <- update(fm4,  ~. -I(watertemp^2))
drop1(fm5, test = "F")

## Single term deletions
##
## Model:
## log10(loss) ~ airflow + watertemp + acidconc + I(airflow^2) +
##     I(acidconc^2) + airflow:watertemp
##                    Df Sum of Sq    RSS  AIC F value Pr(>F)
## <none>                          0.0446 -115
## acidconc            1    0.0022 0.0468 -116    0.69 0.4210
## I(airflow^2)        1    0.0390 0.0836 -104   12.23 0.0036 **
## I(acidconc^2)       1    0.0025 0.0471 -116    0.80 0.3871
## airflow:watertemp   1    0.0194 0.0640 -110    6.10 0.0270 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

fm6 <- update(fm5,  ~. -I(acidconc^2))
drop1(fm6, test = "F")

## Single term deletions
##
## Model:
## log10(loss) ~ airflow + watertemp + acidconc + I(airflow^2) +
##     airflow:watertemp
##                    Df Sum of Sq    RSS  AIC F value Pr(>F)
## <none>                          0.0471 -116
## acidconc           1    0.0050 0.0521 -116    1.58  0.228
## I(airflow^2)       1    0.0437 0.0909 -104   13.92  0.002 **
## airflow:watertemp  1    0.0228 0.0700 -110    7.27  0.017 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

fm7 <- update(fm6,  ~. -acidconc)
drop1(fm7, test = "F")

## Single term deletions
##
## Model:
## log10(loss) ~ airflow + watertemp + I(airflow^2) + airflow:watertemp
##                    Df Sum of Sq    RSS  AIC F value Pr(>F)
## <none>                          0.0521 -116
## I(airflow^2)       1    0.0389 0.0910 -106   11.94 0.0033 **
## airflow:watertemp  1    0.0207 0.0728 -111    6.34 0.0228 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We cannot drop any more variables from the model and the final model is given by:

```
summary(fm7)

##
## Call:
## lm(formula = log10(loss) ~ airflow + watertemp + I(airflow^2) +
##     airflow:watertemp, data = process)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.06438 -0.04206 -0.00639  0.04318  0.11023
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.04669    0.68521   -1.53  0.14615
## airflow            0.10293    0.02280    4.51  0.00035 ***
## watertemp         -0.15909    0.07521   -2.12  0.05045 .
## I(airflow^2)      -0.00124    0.00036   -3.45  0.00326 **
## airflow:watertemp  0.00310    0.00123    2.52  0.02281 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

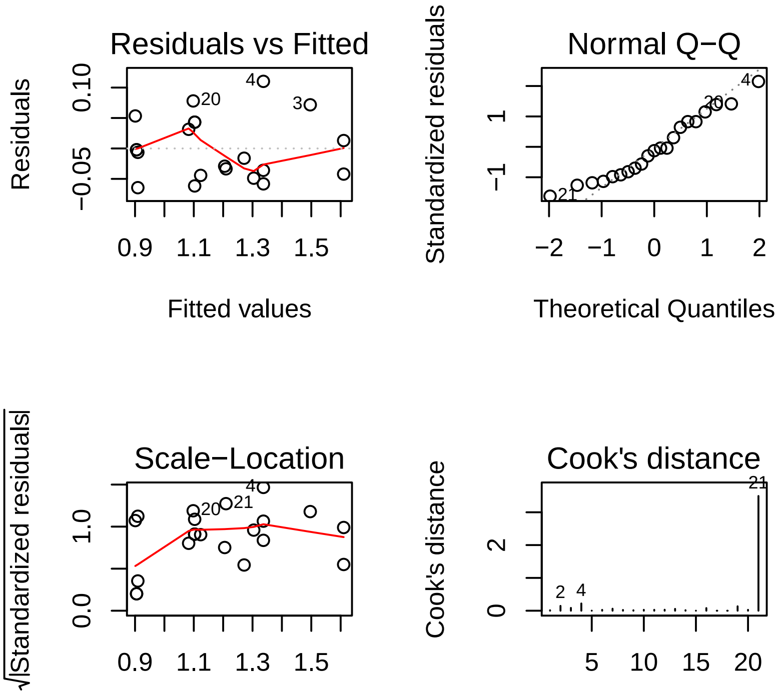
```
##
## Residual standard error: 0.0571 on 16 degrees of freedom
## Multiple R-squared:  0.95,Adjusted R-squared:  0.937
## F-statistic: 75.4 on 4 and 16 DF,  p-value: 3.58e-10

confint(fm7)

##                         2.5 %      97.5 %
## (Intercept)       -2.49927183  0.40590072
## airflow            0.05458804  0.15126650
## watertemp         -0.31853030  0.00035043
## I(airflow^2)      -0.00200901 -0.00048105
## airflow:watertemp  0.00048979  0.00570130

par(mfrow=c(2, 2))
plot(fm7, which=1:4)
par(mfrow=c(1, 1))
```

The process loss depends on airflow and watertemp in a rather complex manner through airflow squared and the interaction between airflow and watertemp. From Figure 5 the plot of Cook's distance we see that observation 21 is an influential observation.

The mathematical formula for the final model is:

$$log10(y_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i1}^2 + \beta_4 x_{i1} \cdot x_{i2} + \epsilon_i$$

Where y_i is the observed loss in observation i, x_{i1} is the airflow for observation i and x_{i2} is the water temperature. We also assume that $\epsilon_i \sim N(0, \sigma^2)$ and all ϵ_i are independent $i = 1, \ldots, 21$.



Figur 5: Model check for final model

2 Extra Analyses

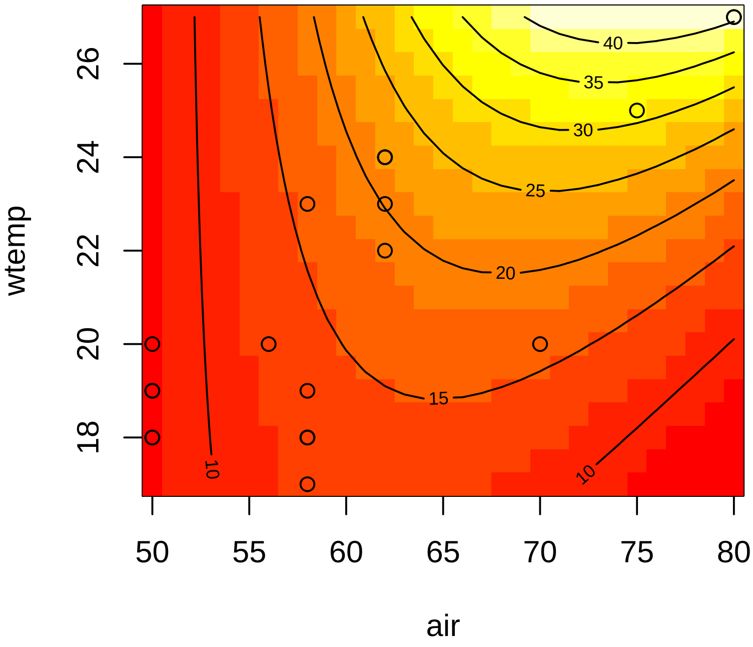
2.1 Plotting the results

The final model had a rather complex dependency of log-loss on watertemp and airflow. to show the results one might want to have a plot of the predicted loss.

In Figure 6 we have a contour plot of the effect of airflow and watertemp on loss. We see that high airflow and high watertemp leads to high loss but the effect is not linear.

```r
## Creating a grid for interpolation
air <- 50:80
wtemp <- seq(17,27,0.5)
pred.data <- expand.grid(airflow=air, watertemp=wtemp)
## Then predicting each grid point,
# taking the values back to the original scale:
```

```
pred.data$fit <- 10^predict(fm7, newdata = pred.data)
## Wrapping my predictions as a matrix with
## length(air) rows and length(wtemp) columns
z <- matrix(pred.data$fit, nrow = length(air))

image(air,wtemp,z)
contour(air,wtemp,z,add=TRUE)
points(watertemp ~ airflow, data = process) # To show the observations
```



Figur 6: Interaction plot for the effect of airflow and watertemp on loss

2.2 Removing an influential observation

In Figure 5 we saw that observation 21 was an influential observation. One couls try to take this observation out to see how much the results change:

```
fm8 <- update(fm7, subset=-21)
summary(fm8)

## 
## Call:
## lm(formula = log10(loss) ~ airflow + watertemp + I(airflow^2) + 
##     airflow:watertemp, data = process, subset = -21)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.06982 -0.03571 -0.00627  0.02374  0.12843 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)  
## (Intercept)       -2.619692   1.121429   -2.34    0.034 *
## airflow            0.068065   0.029591    2.30    0.036 *
## watertemp          0.085314   0.159055    0.54    0.600  
## I(airflow^2)      -0.000177   0.000709   -0.25    0.806  
## airflow:watertemp -0.001107   0.002708   -0.41    0.689  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.0539 on 15 degrees of freedom
## Multiple R-squared:  0.958,Adjusted R-squared:  0.947 
## F-statistic: 85.3 on 4 and 15 DF,  p-value: 3.95e-10
```

Taking observation 21 out did change the model, as now the interaction and also the squared term are no longer significant. If we can find a reason to leave out observation 21 (maybe there was a mistake in the data) then we can reduce the model a bit more:

```
fm9 <- update(fm8, ~. -I(airflow^2))

drop1(fm9, test = "F")

## Single term deletions
## 
## Model:
## log10(loss) ~ airflow + watertemp + airflow:watertemp
##                   Df Sum of Sq    RSS  AIC F value Pr(>F)   
## <none>                         0.0437 -114                  
## airflow:watertemp  1    0.0354 0.0792 -105      13 0.0024 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(fm9)

## 
## Call:
## lm(formula = log10(loss) ~ airflow + watertemp + airflow:watertemp, 
##     data = process, subset = -21)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.06960 -0.03410 -0.00817  0.02526  0.13168 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       -2.835035   0.698235   -4.06  0.00091 ***
## airflow            0.061347   0.012112    5.07  0.00011 ***
## watertemp          0.124384   0.029891    4.16  0.00074 ***
## airflow:watertemp -0.001773   0.000492   -3.60  0.00239 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.0523 on 16 degrees of freedom
## Multiple R-squared:  0.958,Adjusted R-squared:  0.95 
## F-statistic:  121 on 3 and 16 DF,  p-value: 3.36e-11

confint(fm9)

##                        2.5 %      97.5 %
## (Intercept)       -4.3152278 -1.35484278
## airflow            0.0356713  0.08702276
## watertemp          0.0610181  0.18774888
## airflow:watertemp -0.0028168 -0.00072923
```

So now the log-precess loss depends on airflow, watertemp and the interaction between airflow and watertemp.

3 Appendix: R code

```r

setwd("H:/Teaching/PhD_applied_statistics_J16/MultipleRegression/Data")

process <- read.delim("process.txt")
str(process)
library(car)
scatterplotMatrix(process, diagonal= "boxplot")

fm.air <- lm(loss ~ airflow, data = process)
fm.temp <- lm(loss ~ watertemp, data = process)
fm.acid <- lm(loss ~ acidconc, data = process)
summary(fm.air)
summary(fm.temp)
summary(fm.acid)

fm.all <- lm(loss ~ airflow + watertemp + acidconc, data = process)
summary(fm.all)

drop1(fm.all, test = "F")
par(mfrow=c(2, 2))
plot(fm.all, which=1:4)
par(mfrow=c(1, 1))

fm3 <- lm(log10(loss) ~ airflow + watertemp + acidconc, data = process)
par(mfrow=c(2, 2))
plot(fm3, which=1:4)
par(mfrow=c(1, 1))

library(tree)
model<-tree(log10(loss) ~., data = process)
plot(model)
text(model)

fm4 <- lm(log10(loss) ~ airflow + watertemp + acidconc + I(airflow^2) +
            I(watertemp^2) + I(acidconc^2) + airflow:watertemp,
          data = process)

drop1(fm4, test = "F")
fm5 <- update(fm4, ~. -I(watertemp^2))
drop1(fm5, test = "F")

fm6 <- update(fm5, ~. -I(acidconc^2))
```

```
drop1(fm6, test = "F")

fm7 ← update(fm6, ~. -acidconc)
drop1(fm7, test = "F")

summary(fm7)
confint(fm7)

par(mfrow=c(2, 2))
plot(fm7, which=1:4)
par(mfrow=c(1, 1))
## Creating a grid for interpolation
air ← 50:80
wtemp ← seq(17,27,0.5)
pred.data ← expand.grid(airflow=air, watertemp=wtemp)
## Then predicting each grid point,
# taking the values back to the original scale:
pred.data$fit ← 10^predict(fm7, newdata = pred.data)
## Wrapping my predictions as a matrix with
## length(air) rows and length(wtemp) columns
z ← matrix(pred.data$fit, nrow = length(air))

image(air,wtemp,z)
contour(air,wtemp,z,add=TRUE)
points(watertemp ~ airflow, data = process) # To show the observations


fm8 ← update(fm7, subset=-21)
summary(fm8)

fm9 ← update(fm8, ~. -I(airflow^2))

drop1(fm9, test = "F")
summary(fm9)
confint(fm9)
```