AdvStDaAn, Worksheet, Week 2

Micheal Lappert

05.04.2022

Contents

Exercise 1

```
path <- file.path('Datasets', 'sniffer.dat')
df <- read.table(path, header=TRUE)

summary(df)
```

Dataset loading and sanity check:

```
##    Temp.Tank       Temp.Gas       Vapor.Tank    Vapor.Dispensed
##  Min.   :31.00   Min.   :35.00   Min.   :2.590   Min.   :2.590
##  1st Qu.:37.00   1st Qu.:41.00   1st Qu.:3.290   1st Qu.:3.373
##  Median :60.00   Median :60.00   Median :4.285   Median :4.090
##  Mean   :57.91   Mean   :55.91   Mean   :4.422   Mean   :4.324
##  3rd Qu.:62.00   3rd Qu.:62.00   3rd Qu.:4.630   3rd Qu.:4.540
##  Max.   :92.00   Max.   :92.00   Max.   :7.450   Max.   :7.450
##        Y
##  Min.   :16.00
##  1st Qu.:23.75
##  Median :31.50
##  Mean   :31.12
##  3rd Qu.:34.50
##  Max.   :55.00
```

```
dim(df)
```

```
## [1] 32  5
```

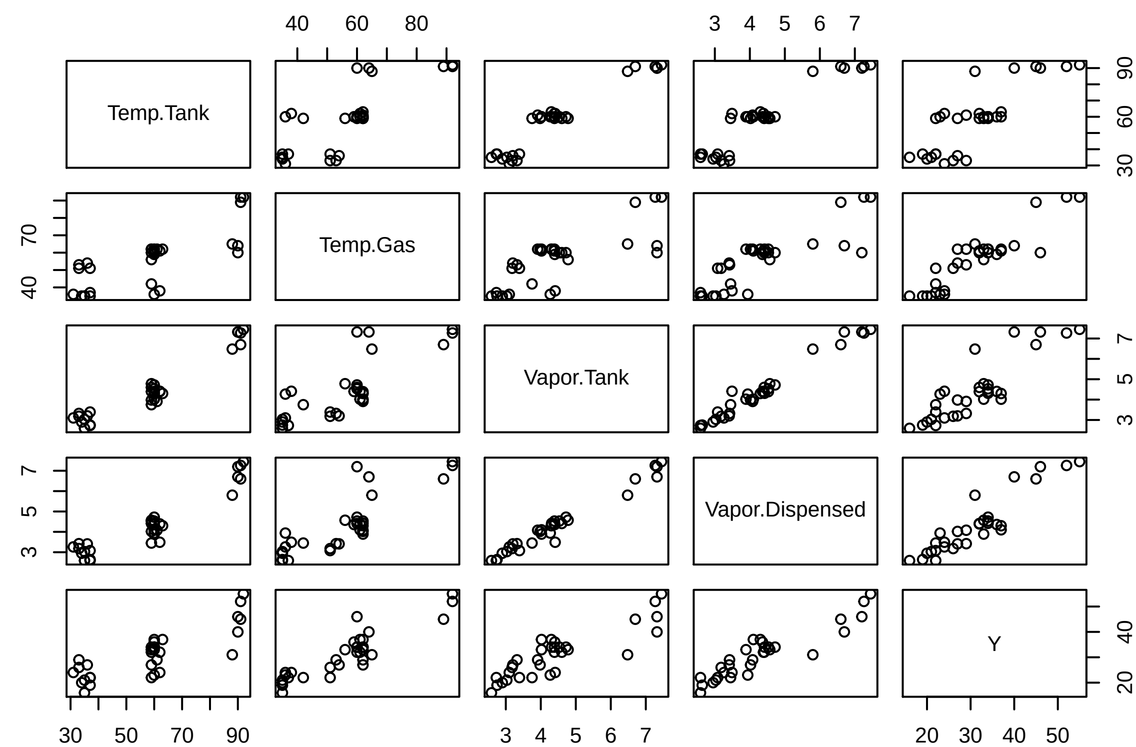
```r
head(df)
```

```
##   Temp.Tank Temp.Gas Vapor.Tank Vapor.Dispensed  Y
## 1        33       53       3.32            3.42 29
## 2        31       36       3.10            3.26 24
## 3        33       51       3.18            3.18 26
## 4        37       51       3.39            3.08 22
## 5        36       54       3.20            3.41 27
## 6        35       35       3.03            3.03 21
```

```r
tail(df)
```

```
##    Temp.Tank Temp.Gas Vapor.Tank Vapor.Dispensed  Y
## 27        60       62       4.02            3.89 33
## 28        59       62       3.98            4.02 27
## 29        59       62       4.39            4.53 34
## 30        37       35       2.75            2.64 19
## 31        35       35       2.59            2.59 16
## 32        37       37       2.73            2.59 22
```

```r
plot(df)
```



Data looks like it is highly correlated with each other. But we keep it this way for the first exercises.

Exercise 1.a)

Fitting a first model without any transformations to the data:

```r
lm1.1 <- lm(Y ~ ., data = df)
```

The model looks initially not too bad. For a proper evaluation one would need to perform a residual and sensitivity analysis to investigate the adequacy of the model. But for this exercise we keep the track of the worksheet.

E1.a)(I) Estimated coefficients

```r
coef(lm1.1)
```

```
##     (Intercept)       Temp.Tank        Temp.Gas      Vapor.Tank Vapor.Dispensed
##      1.01501756     -0.02860886      0.21581693     -4.32005167      8.97488928
```

E1.a)(II) F-statistic

```r
summary(lm1.1)
```

```
##
## Call:
## lm(formula = Y ~ ., data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.586 -1.221 -0.118  1.320  5.106
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.01502    1.86131   0.545  0.59001
## Temp.Tank       -0.02861    0.09060  -0.316  0.75461
## Temp.Gas         0.21582    0.06772   3.187  0.00362 **
## Vapor.Tank      -4.32005    2.85097  -1.515  0.14132
## Vapor.Dispensed  8.97489    2.77263   3.237  0.00319 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.73 on 27 degrees of freedom
## Multiple R-squared:  0.9261, Adjusted R-squared:  0.9151
## F-statistic: 84.54 on 4 and 27 DF,  p-value: 7.249e-15
```

The p-value of the F-statistic is « 0.05 indicating that at least one of the variables can not be 0 and therfore are important to describe the response value. Even though, the p-values of the t-test indicate that not all of them are of the same importance. In this case are only 2 explanatory variables significantly important (Temp.Gas & Vapor.Dispensed).

E1.a)(III) Variance Inflation Factor (VIF)
Inspecting multicollinearity with the Variance Inflation Factor (VIF):

```r
library(car)
```

```
## Loading required package: carData
```

```r
vif(lm1.1)
```

```
##       Temp.Tank        Temp.Gas      Vapor.Tank Vapor.Dispensed
##       12.997379        4.720998       71.301491       61.932647
```

A vif above 5 to 10 indicates problems with multicollinearity. According to this guideline all variables but Temp.Gas have too high vif factors and therewith problems with multicollinearity. Vapor.Tank is affected the most.

Exercise 1.b)

Performing a variable selection using the AIC stepwise from the model fitted in Exercise 1.a):

```r
step(lm1.1)
```

```
## Start:  AIC=68.84
## Y ~ Temp.Tank + Temp.Gas + Vapor.Tank + Vapor.Dispensed
##
##                   Df Sum of Sq    RSS    AIC
## - Temp.Tank        1     0.743 201.97 66.956
## <none>                         201.23 68.838
## - Vapor.Tank       1    17.113 218.34 69.450
## - Temp.Gas         1    75.698 276.93 77.056
## - Vapor.Dispensed  1    78.090 279.32 77.332
##
## Step:  AIC=66.96
## Y ~ Temp.Gas + Vapor.Tank + Vapor.Dispensed
##
##                   Df Sum of Sq    RSS    AIC
## <none>                         201.97 66.956
## - Vapor.Tank       1    36.416 238.39 70.261
## - Temp.Gas         1    78.831 280.80 75.501
## - Vapor.Dispensed  1    91.850 293.82 76.952
```

```
##
## Call:
## lm(formula = Y ~ Temp.Gas + Vapor.Tank + Vapor.Dispensed, data = df)
##
## Coefficients:
##     (Intercept)         Temp.Gas       Vapor.Tank  Vapor.Dispensed
##          1.0655           0.2091          -4.8882           9.2480
```

The best model with the stepwise variable selection from the model in Exercise 1.a) is
Y ~ Temp.Gas + Vapor.Tank + Vapor.Dispensed
Temp.Tank gets not included. This would be due to multicollinearity with other variables.

Exercise 1.c)

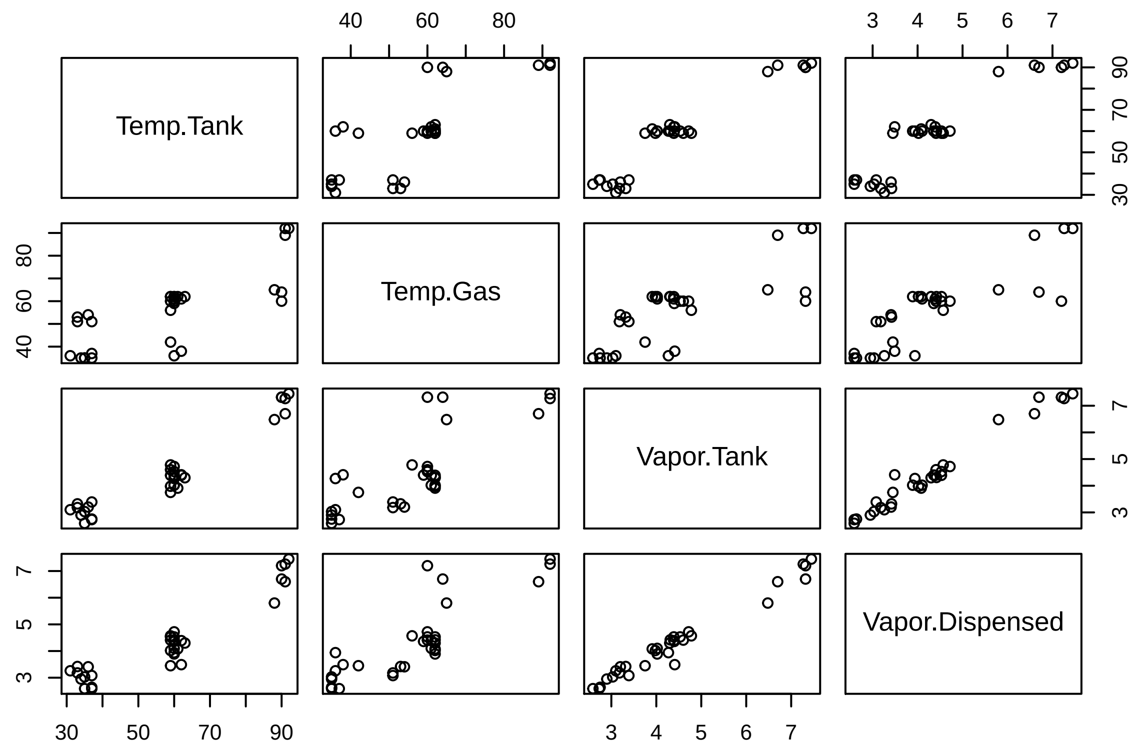
Did we already remedy the initially found multicollinearity with the stepwise variable selection? We can check by performing a vif on the newly found model.

```
lm1.2 <- lm(Y ~ Temp.Gas + Vapor.Tank + Vapor.Dispensed, data = df)
vif(lm1.2)
```

```
##        Temp.Gas      Vapor.Tank Vapor.Dispensed
##        4.255787       42.899447       55.907555
```

No, Vapor.Tank and Vapor.Dispensed have still vif values from above 5 to 10. Which ones are correlated the most?

```
pairs(df[,-5])
```



Vapor.Tank and Vapor.Dispensed seem to be correlated the most. So we try transformations of the variables by replacing them by the mean and the difference.

```
df2 <- data.frame(diffVapor = df$Vapor.Tank - df$Vapor.Dispensed,
                  meanVapor = (df$Vapor.Tank + df$Vapor.Dispensed) / 2)

df3 <- cbind(df2, Temp.Tank = df$Temp.Tank, Temp.Gas = df$Temp.Gas, Y = df$Y)

head(df3)
```

```
##   diffVapor meanVapor Temp.Tank Temp.Gas  Y
## 1     -0.10     3.370        33       53 29
## 2     -0.16     3.180        31       36 24
## 3      0.00     3.180        33       51 26
## 4      0.31     3.235        37       51 22
## 5     -0.21     3.305        36       54 27
## 6      0.00     3.030        35       35 21
```

With the newly created data.frame with the transformed variables one can now perform another stepwise variable selection.

```r
lm1.3 <- lm(Y ~ ., data = df3)
step(lm1.3)
```

```
## Start:  AIC=68.84
## Y ~ diffVapor + meanVapor + Temp.Tank + Temp.Gas
##
##             Df Sum of Sq    RSS    AIC
## - Temp.Tank  1     0.743 201.97 66.956
## <none>                   201.23 68.838
## - diffVapor  1    43.585 244.81 73.112
## - Temp.Gas   1    75.698 276.93 77.056
## - meanVapor  1   114.810 316.04 81.284
##
## Step:  AIC=66.96
## Y ~ diffVapor + meanVapor + Temp.Gas
##
##             Df Sum of Sq    RSS    AIC
## <none>                   201.97 66.956
## - diffVapor  1    64.398 266.37 73.813
## - Temp.Gas   1    78.831 280.80 75.501
## - meanVapor  1   265.710 467.68 91.826

##
## Call:
## lm(formula = Y ~ diffVapor + meanVapor + Temp.Gas, data = df3)
##
## Coefficients:
## (Intercept)    diffVapor    meanVapor     Temp.Gas
##      1.0655      -7.0681       4.3597       0.2091
```

This is the same model as found in Exercise 1.b) but with the transformed variables. Now one can check if the problems with multicollinearity still persists.

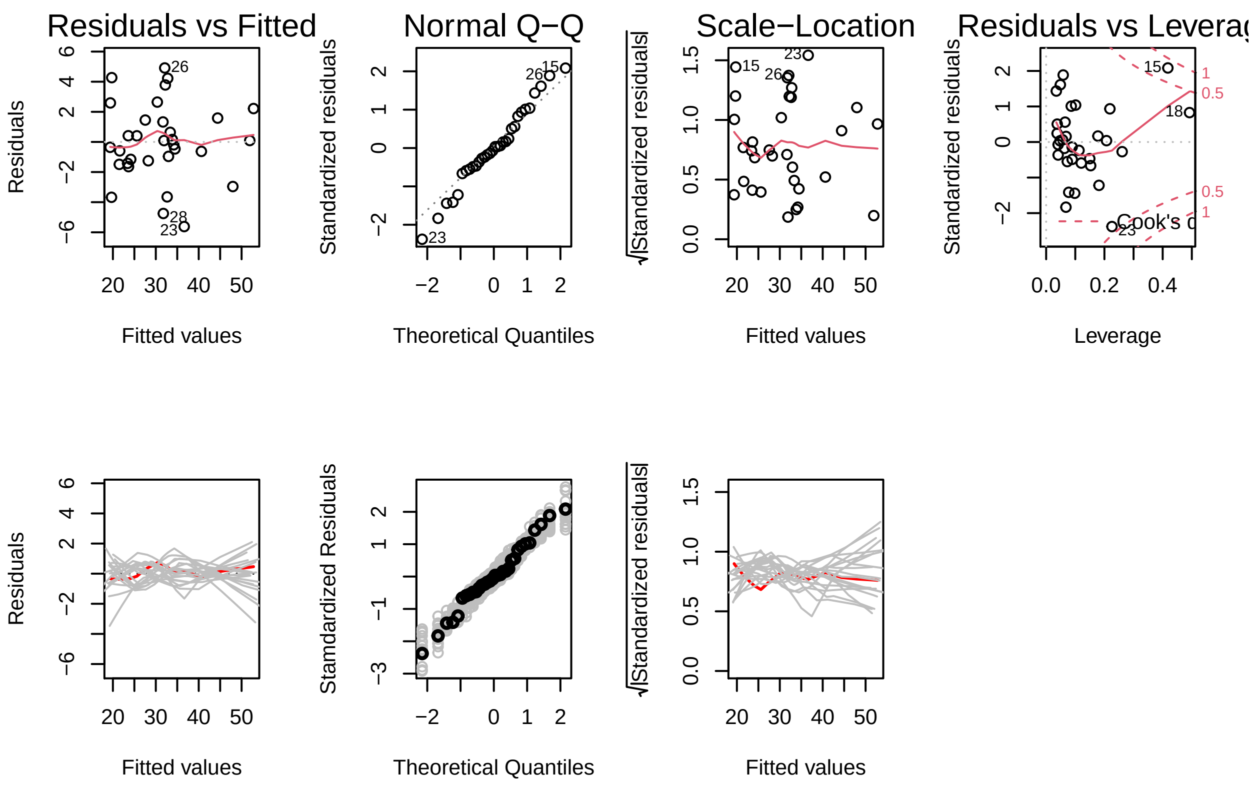
```r
lm1.4 <- lm(Y ~ diffVapor + meanVapor + Temp.Gas, data = df3)
vif(lm1.4)
```

```
## diffVapor meanVapor  Temp.Gas
##  1.538981  4.450470  4.255787
```

All vif values are lower than 5 and therewith the problem with multicollinearity does not persist.

How looks the residual and sensitivity analysis?

```
par(mfrow = c(2, 4))
plot(lm1.4)
plot.lmSim(lm1.4, SEED = 1)
```



leverage points > 0.25

There is no evidence that any of the assumptions is violated.