Task 22

Lisa Skalon

5/1/2020

```
library('ggplot2')
library('ggpubr')
library('dplyr')
library('tidyr')
library('broom')
library('lubridate')
library('reshape2')
library('car')
```

Airquality - part 2

We read df again and clean it almost the same as in the part 1.

```
# read
df <- read.csv("./AirQualityUCI.csv", stringsAsFactors = FALSE, sep = ';')

# examine the data structure
str(df)
```

```
## 'data.frame':    9471 obs. of  17 variables:
##  $ Date         : chr  "10/03/2004" "10/03/2004" "10/03/2004" "10/03/2004" ...
##  $ Time         : chr  "18.00.00" "19.00.00" "20.00.00" "21.00.00" ...
##  $ CO.GT.       : chr  "2,6" "2" "2,2" "2,2" ...
##  $ PT08.S1.CO.  : int  1360 1292 1402 1376 1272 1197 1185 1136 1094 1010 ...
##  $ NMHC.GT.     : int  150 112 88 80 51 38 31 31 24 19 ...
##  $ C6H6.GT.     : chr  "11,9" "9,4" "9,0" "9,2" ...
##  $ PT08.S2.NMHC.: int  1046 955 939 948 836 750 690 672 609 561 ...
##  $ NOx.GT.      : int  166 103 131 172 131 89 62 62 45 -200 ...
##  $ PT08.S3.NOx. : int  1056 1174 1140 1092 1205 1337 1462 1453 1579 1705 ...
##  $ NO2.GT.      : int  113 92 114 122 116 96 77 76 60 -200 ...
##  $ PT08.S4.NO2. : int  1692 1559 1555 1584 1490 1393 1333 1333 1276 1235 ...
##  $ PT08.S5.O3.  : int  1268 972 1074 1203 1110 949 733 730 620 501 ...
##  $ T            : chr  "13,6" "13,3" "11,9" "11,0" ...
##  $ RH           : chr  "48,9" "47,7" "54,0" "60,0" ...
##  $ AH           : chr  "0,7578" "0,7255" "0,7502" "0,7867" ...
##  $ X            : logi  NA NA NA NA NA NA ...
##  $ X.1          : logi  NA NA NA NA NA NA ...
```

```
# convert date and time to special type
df$date_time = dmy_hms(paste(df$Date, df$Time))

# remove unuseful cols
df <- df[,c(3:15,18) ]
```

```r
# chr to numeric
char_columns <- sapply(df, is.character)
df[ , char_columns] <- lapply(df[ , char_columns] , function(x) as.numeric(gsub(",", ".", x)))

# remove na
df <- na.omit(df)

# value -200 is suspicious
# let values -200 be NA
df_no200 <- df[, 1:13]
df_no200[] <- sapply(df[, 1:13] , function(x) {x[grep("-200", x)] = NA; return((x))})
str(df_no200)
```

```
## 'data.frame':    9357 obs. of  13 variables:
##  $ CO.GT.       : num  2.6 2 2.2 2.2 1.6 1.2 1.2 1 0.9 0.6 ...
##  $ PT08.S1.CO.  : num  1360 1292 1402 1376 1272 ...
##  $ NMHC.GT.     : num  150 112 88 80 51 38 31 31 24 19 ...
##  $ C6H6.GT.     : num  11.9 9.4 9 9.2 6.5 4.7 3.6 3.3 2.3 1.7 ...
##  $ PT08.S2.NMHC.: num  1046 955 939 948 836 ...
##  $ NOx.GT.      : num  166 103 131 172 131 89 62 62 45 NA ...
##  $ PT08.S3.NOx. : num  1056 1174 1140 1092 1205 ...
##  $ NO2.GT.      : num  113 92 114 122 116 96 77 76 60 NA ...
##  $ PT08.S4.NO2. : num  1692 1559 1555 1584 1490 ...
##  $ PT08.S5.O3.  : num  1268 972 1074 1203 1110 ...
##  $ T            : num  13.6 13.3 11.9 11 11.2 11.2 11.3 10.7 10.7 10.3 ...
##  $ RH           : num  48.9 47.7 54 60 59.6 59.2 56.8 60 59.7 60.2 ...
##  $ AH           : num  0.758 0.726 0.75 0.787 0.789 ...
```

```r
summary(df_no200)
```

```
##      CO.GT.         PT08.S1.CO.      NMHC.GT.         C6H6.GT.
##  Min.   : 0.100   Min.   : 647   Min.   :   7.0   Min.   : 0.10
##  1st Qu.: 1.100   1st Qu.: 937   1st Qu.:  67.0   1st Qu.: 4.40
##  Median : 1.800   Median :1063   Median : 150.0   Median : 8.20
##  Mean   : 2.153   Mean   :1100   Mean   : 218.8   Mean   :10.08
##  3rd Qu.: 2.900   3rd Qu.:1231   3rd Qu.: 297.0   3rd Qu.:14.00
##  Max.   :11.900   Max.   :2040   Max.   :1189.0   Max.   :63.70
##  NA's   :1683     NA's   :366    NA's   :8443     NA's   :366
##  PT08.S2.NMHC.       NOx.GT.        PT08.S3.NOx.       NO2.GT.
##  Min.   : 383.0   Min.   :   2.0   Min.   : 322.0   Min.   :  2.0
##  1st Qu.: 734.5   1st Qu.:  98.0   1st Qu.: 658.0   1st Qu.: 78.0
##  Median : 909.0   Median : 180.0   Median : 806.0   Median :109.0
##  Mean   : 939.2   Mean   : 246.9   Mean   : 835.5   Mean   :113.1
##  3rd Qu.:1116.0   3rd Qu.: 326.0   3rd Qu.: 969.5   3rd Qu.:142.0
##  Max.   :2214.0   Max.   :1479.0   Max.   :2683.0   Max.   :340.0
##  NA's   :366      NA's   :1639     NA's   :366      NA's   :1642
##   PT08.S4.NO2.   PT08.S5.O3.           T               RH
##  Min.   : 551   Min.   : 221.0   Min.   :-1.90   Min.   : 9.20
##  1st Qu.:1227   1st Qu.: 731.5   1st Qu.:11.80   1st Qu.:35.80
##  Median :1463   Median : 963.0   Median :17.80   Median :49.60
##  Mean   :1456   Mean   :1022.9   Mean   :18.32   Mean   :49.23
##  3rd Qu.:1674   3rd Qu.:1273.5   3rd Qu.:24.40   3rd Qu.:62.50
##  Max.   :2775   Max.   :2523.0   Max.   :44.60   Max.   :88.70
##  NA's   :366    NA's   :366      NA's   :366     NA's   :366
```

```
##        AH
##  Min.   :0.1847
##  1st Qu.:0.7368
##  Median :0.9954
##  Mean   :1.0255
##  3rd Qu.:1.3137
##  Max.   :2.2310
##  NA's   :366
```

```
# We remove almost all suspicious values, but the column NMHC.GT. contains too many of them,
# so for that column we replace -200 with median, because we don`t want to loose information
df_no200 <- df_no200[!(is.na(df_no200$AH) | is.na(df_no200$NO2.GT.) | is.na(df_no200$CO.GT.)),]

df_narm <- replace(df_no200, TRUE, lapply(df_no200, function(x) replace(x, is.na(x), median(x, na.rm = 

summary(df_narm)
```

```
##      CO.GT.         PT08.S1.CO.      NMHC.GT.         C6H6.GT.
##  Min.   : 0.100   Min.   : 647   Min.   :   7.0   Min.   : 0.20
##  1st Qu.: 1.100   1st Qu.: 956   1st Qu.: 157.0   1st Qu.: 4.90
##  Median : 1.900   Median :1085   Median : 157.0   Median : 8.80
##  Mean   : 2.182   Mean   :1120   Mean   : 165.8   Mean   :10.55
##  3rd Qu.: 2.900   3rd Qu.:1254   3rd Qu.: 157.0   3rd Qu.:14.60
##  Max.   :11.900   Max.   :2040   Max.   :1189.0   Max.   :63.70
##  PT08.S2.NMHC.       NOx.GT.        PT08.S3.NOx.       NO2.GT.
##  Min.   : 390.0   Min.   :   2.0   Min.   : 322.0   Min.   :  2.0
##  1st Qu.: 760.0   1st Qu.: 103.0   1st Qu.: 642.0   1st Qu.: 79.0
##  Median : 931.0   Median : 186.0   Median : 786.0   Median :110.0
##  Mean   : 958.5   Mean   : 250.7   Mean   : 816.9   Mean   :113.9
##  3rd Qu.:1135.0   3rd Qu.: 335.0   3rd Qu.: 947.0   3rd Qu.:142.0
##  Max.   :2214.0   Max.   :1479.0   Max.   :2683.0   Max.   :333.0
##   PT08.S4.NO2.    PT08.S5.O3.          T                RH              AH
##  Min.   : 551   Min.   : 221   Min.   :-1.90   Min.   : 9.20   Min.   :0.1847
##  1st Qu.:1207   1st Qu.: 760   1st Qu.:11.20   1st Qu.:35.30   1st Qu.:0.6941
##  Median :1457   Median :1006   Median :16.80   Median :49.20   Median :0.9539
##  Mean   :1453   Mean   :1058   Mean   :17.76   Mean   :48.88   Mean   :0.9856
##  3rd Qu.:1683   3rd Qu.:1322   3rd Qu.:23.70   3rd Qu.:62.20   3rd Qu.:1.2516
##  Max.   :2775   Max.   :2523   Max.   :44.60   Max.   :88.70   Max.   :2.1806
```

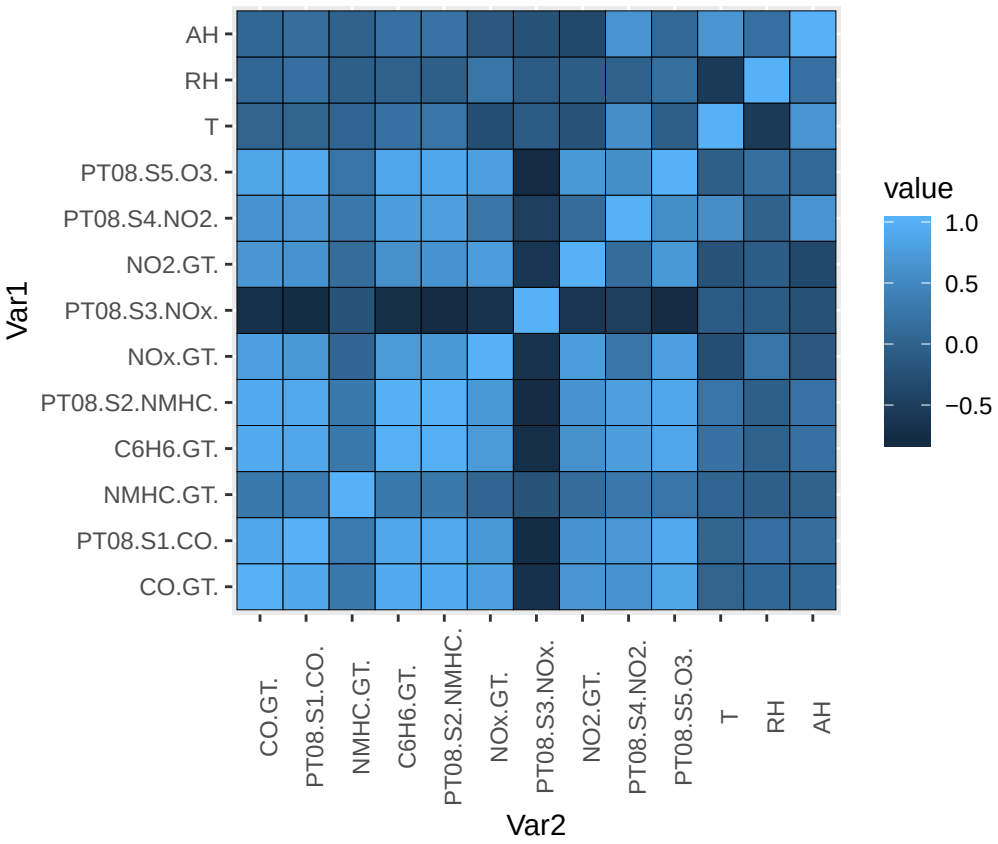
For exploring collinearity of predictors we can simple check for cross-correlations between variables. If the correlation is greater than 0.8/lower than -0.8, probably these predictors depend on each other and should be removed from the analysis.

```
# correlation heatmap - more comportable way to find cross-correlations
cor_mtx <- (cor(df_narm))
cor_mtx
```

```
##                    CO.GT. PT08.S1.CO.     NMHC.GT.    C6H6.GT. PT08.S2.NMHC.
## CO.GT.          1.00000000  0.87701378  0.297783114  0.93000844    0.91431022
## PT08.S1.CO.     0.87701378  1.00000000  0.329639764  0.87743017    0.88606831
## NMHC.GT.        0.29778311  0.32963976  1.000000000  0.29539169    0.29229243
## C6H6.GT.        0.93000844  0.87743017  0.295391689  1.00000000    0.98270456
## PT08.S2.NMHC.   0.91431022  0.88606831  0.292292432  0.98270456    1.00000000
## NOx.GT.         0.78645556  0.70770473  0.044360592  0.71834383    0.70535922
## PT08.S3.NOx.   -0.70103781 -0.76289451 -0.199860844 -0.72572179   -0.78163029
## NO2.GT.         0.67383957  0.62826276  0.124697881  0.60324096    0.63330954
```

```
## PT08.S4.NO2.   0.63083404  0.67590960  0.286329429  0.76180525    0.77428754
## PT08.S5.O3.    0.85348004  0.89716636  0.242022839  0.86115359    0.87677728
## T              0.01833425  0.02827677  0.037186468  0.18900256    0.22833308
## RH             0.06475315  0.16923408 -0.044807229 -0.02159153   -0.04608374
## AH             0.05934617  0.14975239 -0.003499192  0.18707157    0.20559028
##                   NOx.GT. PT08.S3.NOx.    NO2.GT. PT08.S4.NO2. PT08.S5.O3.
## CO.GT.         0.78645556  -0.70103781  0.6738396  0.630834042  0.85348004
## PT08.S1.CO.    0.70770473  -0.76289451  0.6282628  0.675909601  0.89716636
## NMHC.GT.       0.04436059  -0.19986084  0.1246979  0.286329429  0.24202284
## C6H6.GT.       0.71834383  -0.72572179  0.6032410  0.761805247  0.86115359
## PT08.S2.NMHC.  0.70535922  -0.78163029  0.6333095  0.774287538  0.87677728
## NOx.GT.        1.00000000  -0.66216631  0.7570291  0.233792602  0.78855025
## PT08.S3.NOx.  -0.66216631   1.00000000 -0.6413769 -0.511223442 -0.79336411
## NO2.GT.        0.75702912  -0.64137694  1.0000000  0.142612010  0.70252411
## PT08.S4.NO2.   0.23379260  -0.51122344  0.1426120  1.000000000  0.57424200
## PT08.S5.O3.    0.78855025  -0.79336411  0.7025241  0.574242002  1.00000000
## T             -0.27599835  -0.09949483 -0.2143247  0.566585555 -0.04614601
## RH             0.23225519  -0.11647947 -0.0753329 -0.009160077  0.16482121
## AH            -0.14418603  -0.22338099 -0.3496460  0.646390126  0.07580693
##                          T           RH           AH
## CO.GT.         0.01833425  0.064753147  0.059346169
## PT08.S1.CO.    0.02827677  0.169234080  0.149752388
## NMHC.GT.       0.03718647 -0.044807229 -0.003499192
## C6H6.GT.       0.18900256 -0.021591530  0.187071565
## PT08.S2.NMHC.  0.22833308 -0.046083742  0.205590275
## NOx.GT.       -0.27599835  0.232255190 -0.144186032
## PT08.S3.NOx.  -0.09949483 -0.116479467 -0.223380993
## NO2.GT.       -0.21432470 -0.075332896 -0.349646009
## PT08.S4.NO2.   0.56658556 -0.009160077  0.646390126
## PT08.S5.O3.   -0.04614601  0.164821208  0.075806932
## T              1.00000000 -0.563908917  0.660638059
## RH            -0.56390892  1.000000000  0.179575919
## AH             0.66063806  0.179575919  1.000000000
```

```
# heatmap
ggplot(data = melt(cor_mtx), aes(Var2, Var1, fill = value))+
 geom_tile(color = "black")+
 theme(axis.text.x = element_text(angle = 90))+
  coord_fixed()
```



Then 3 or more predictors depend on each other, we are talking about multicollinearity. It can be measured by VIF score -the variance inflation factor, which evaluate how much the variance of a regression coefficient is inflated due to multicollinearity in the model. We can calculate VIF for each predictor:

```
fit <- lm(data = df_narm, formula =  C6H6.GT. ~ .-NMHC.GT. )
summary(fit)
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ . - NMHC.GT., data = df_narm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3188 -0.6809 -0.1593  0.5259 21.6256
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.613e+01  2.599e-01 -62.068  < 2e-16 ***
## CO.GT.         7.534e-01  2.891e-02  26.057  < 2e-16 ***
## PT08.S1.CO.    2.791e-04  1.759e-04   1.587  0.11257
## PT08.S2.NMHC.  2.461e-02  2.703e-04  91.047  < 2e-16 ***
## NOx.GT.        3.192e-03  1.734e-04  18.404  < 2e-16 ***
## PT08.S3.NOx.   2.579e-03  1.135e-04  22.719  < 2e-16 ***
## NO2.GT.       -1.101e-02  5.797e-04 -18.988  < 2e-16 ***
## PT08.S4.NO2.   1.188e-03  1.582e-04   7.511 6.62e-14 ***
## PT08.S5.O3.   -2.712e-04  9.489e-05  -2.858  0.00427 **
## T             -7.836e-02  5.529e-03 -14.172  < 2e-16 ***
## RH            -2.802e-02  2.148e-03 -13.044  < 2e-16 ***
```

```
## AH             8.174e-01  1.071e-01   7.632 2.62e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.089 on 6929 degrees of freedom
## Multiple R-squared:  0.9788, Adjusted R-squared:  0.9787
## F-statistic: 2.902e+04 on 11 and 6929 DF,  p-value: < 2.2e-16
```

```
vif(fit)
```

```
##        CO.GT.   PT08.S1.CO. PT08.S2.NMHC.       NOx.GT.  PT08.S3.NOx.
##     10.159922      8.662440     29.810356      7.661921      4.783359
##       NO2.GT.  PT08.S4.NO2.   PT08.S5.O3.             T            RH
##      4.433180     18.290829      8.707801     13.994303      8.205014
##            AH
##     10.800525
```

It seeams that many predictors have too big VIF score (>10). These predictors shouldn‘t be included into our model.

Now we can choose predictors which are significant and correlated with our response but have low VIF score. Let‘s try out some of them.

First of all, we check the most significant predictors CO.GT. and PT08.S2.NMHC.

```
fit1 <- lm(data = df_narm, formula =  C6H6.GT. ~  CO.GT.  +  PT08.S2.NMHC.)
summary(fit1)
```

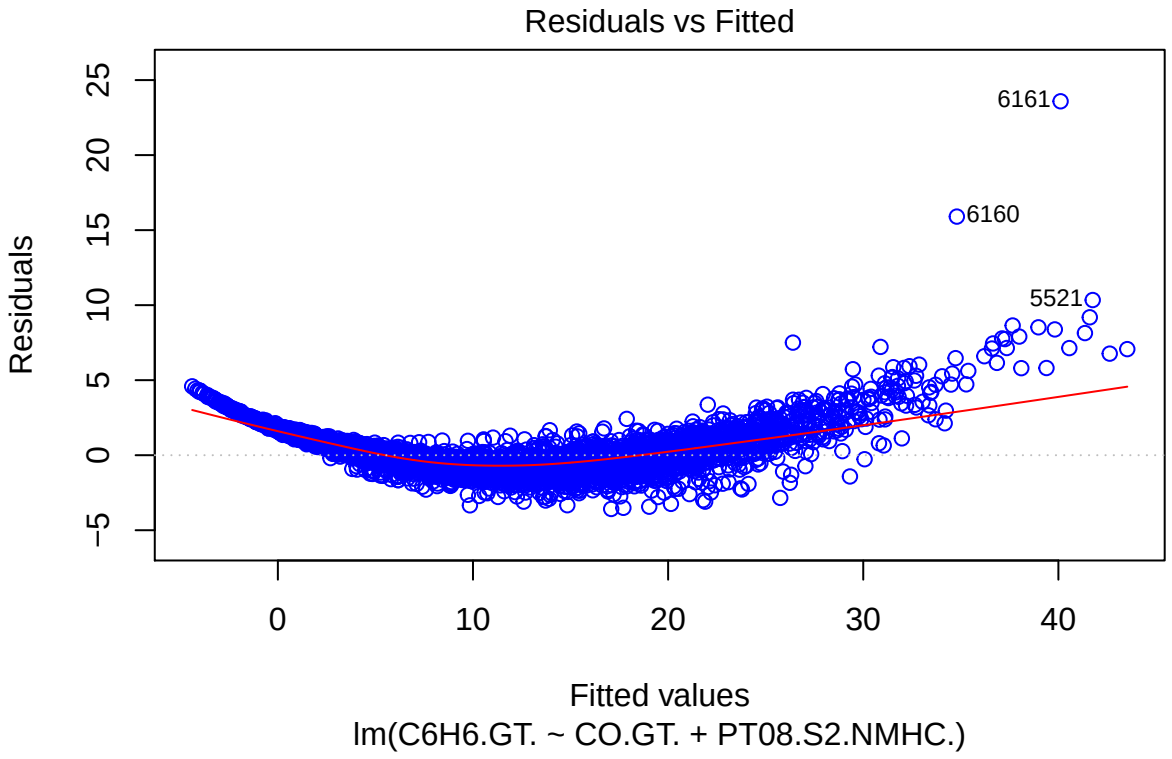
```
##
## Call:
## lm(formula = C6H6.GT. ~ CO.GT. + PT08.S2.NMHC., data = df_narm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5833 -0.7527 -0.2963  0.5342 23.5868
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.349e+01  8.786e-02 -153.52   <2e-16 ***
## CO.GT.         9.951e-01  2.580e-02   38.56   <2e-16 ***
## PT08.S2.NMHC.  2.282e-02  1.408e-04  162.02   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.255 on 6938 degrees of freedom
## Multiple R-squared:  0.9718, Adjusted R-squared:  0.9718
## F-statistic: 1.194e+05 on 2 and 6938 DF,  p-value: < 2.2e-16
```

```
vif(fit1)
```

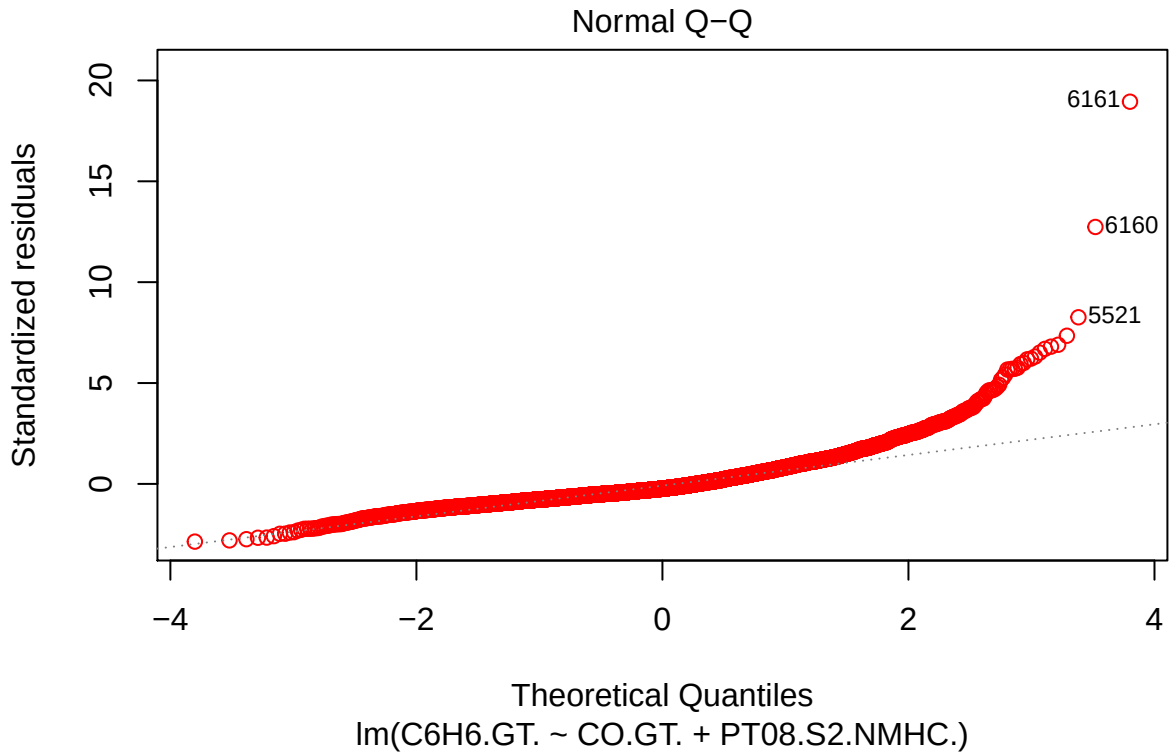
```
##        CO.GT. PT08.S2.NMHC.
##      6.096192      6.096192
```

VIF score for them is not very big, but quantiles are not symmetrical. We can check the residuals of this model: residuals represent variation left unexplained by the model.

```
plot(fit1, which=1, col=c("blue"))
```




```
plot(fit1, which=2, col=c("red"))
```



The residuals doesn't look linear. The are not normally distributed. The model is not good enough.

We will try to choose either CO.GT. or PT08.S2.NMHC. as predictors because they both are very significant but strongly correlated.

Let's try PT08.S5.O3. and RH

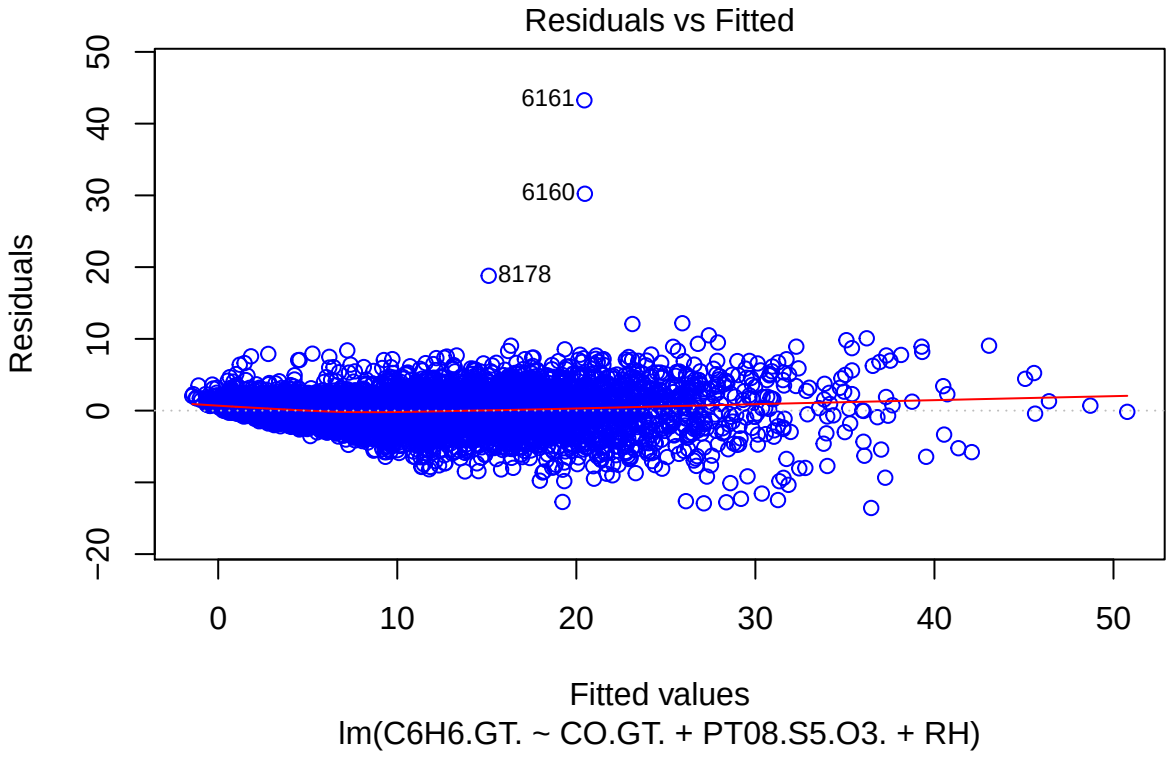
```
fit2 <- lm(data = df_narm, formula =  C6H6.GT. ~  CO.GT.  +  PT08.S5.O3. + RH)
summary(fit2)
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ CO.GT. + PT08.S5.O3. + RH, data = df_narm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.567  -1.265   0.067   1.234  43.251
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.5239359  0.1113979  -4.703 2.61e-06 ***
## CO.GT.       3.5541869  0.0392617  90.526  < 2e-16 ***
## PT08.S5.O3.  0.0054070  0.0001408  38.395  < 2e-16 ***
## RH          -0.0490523  0.0017149 -28.604  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.429 on 6937 degrees of freedom
## Multiple R-squared:  0.8941, Adjusted R-squared:  0.8941
## F-statistic: 1.953e+04 on 3 and 6937 DF,  p-value: < 2.2e-16
```

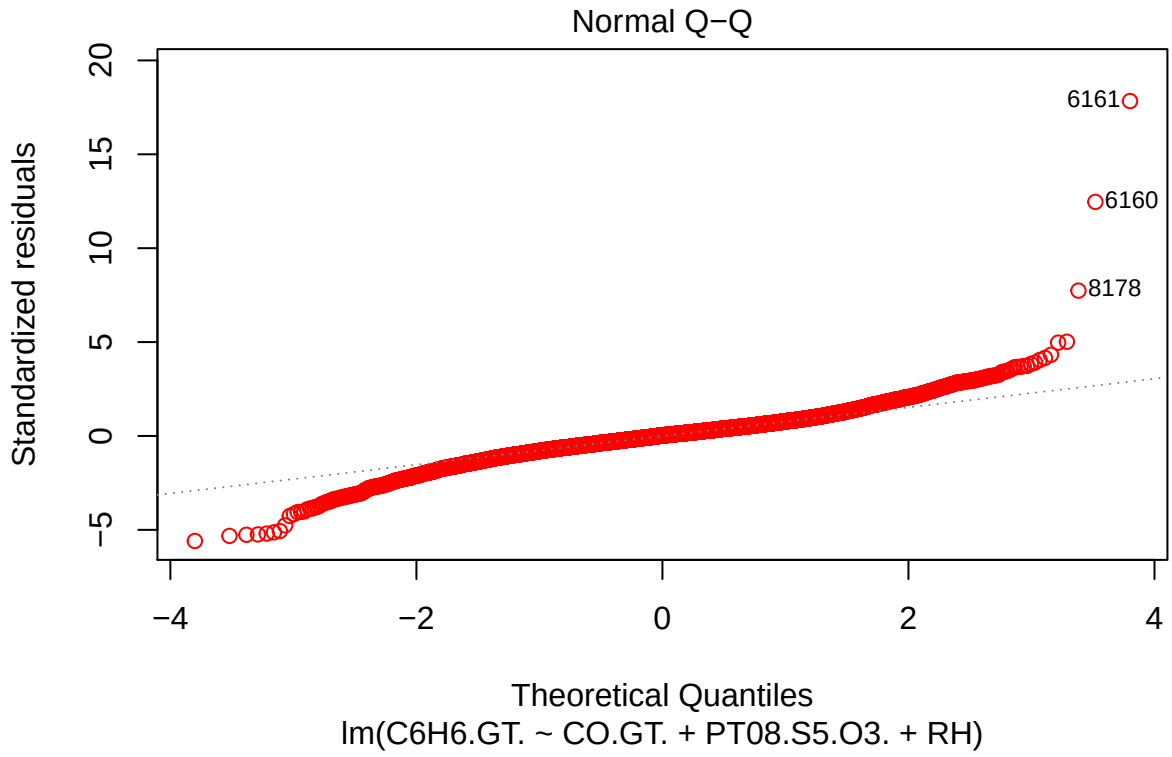
```
vif(fit2)
```

```
##      CO.GT. PT08.S5.O3.          RH
##    3.764390    3.853285    1.050850
```

```
plot(fit2, which=1, col=c("blue"))
```



```
plot(fit2, which=2, col=c("red"))
```



Quantiles are symmetrical, VIF is ok, residuals show linearity and are almost normally distributed.

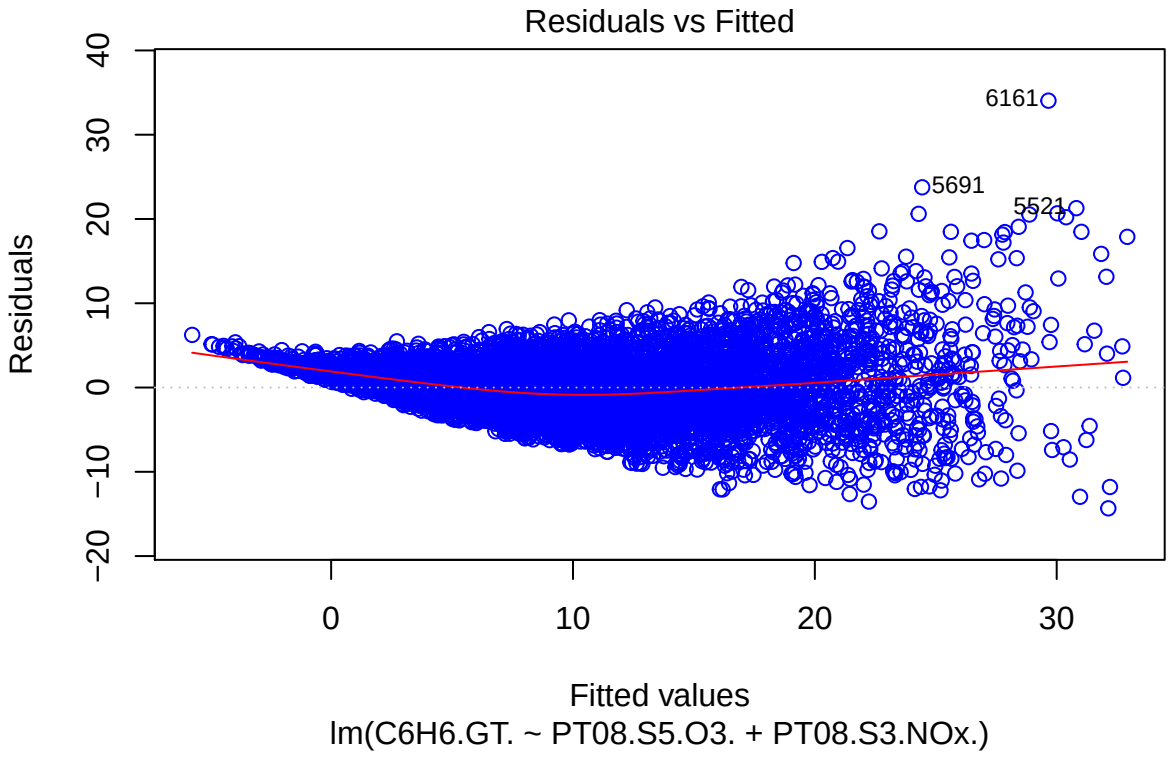
```
fit3 <- lm(data = df_narm, formula =  C6H6.GT. ~  PT08.S5.O3.+ PT08.S3.NOx.)
summary(fit3)
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S5.O3. + PT08.S3.NOx., data = df_narm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.335  -2.324  -0.098   2.177  34.041
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.6278940  0.4130679  -3.941 8.19e-05 ***
## PT08.S5.O3.   0.0141429  0.0001824  77.553  < 2e-16 ***
## PT08.S3.NOx. -0.0033999  0.0002943 -11.553  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.759 on 6938 degrees of freedom
## Multiple R-squared:  0.7465, Adjusted R-squared:  0.7464
## F-statistic: 1.021e+04 on 2 and 6938 DF,  p-value: < 2.2e-16
```

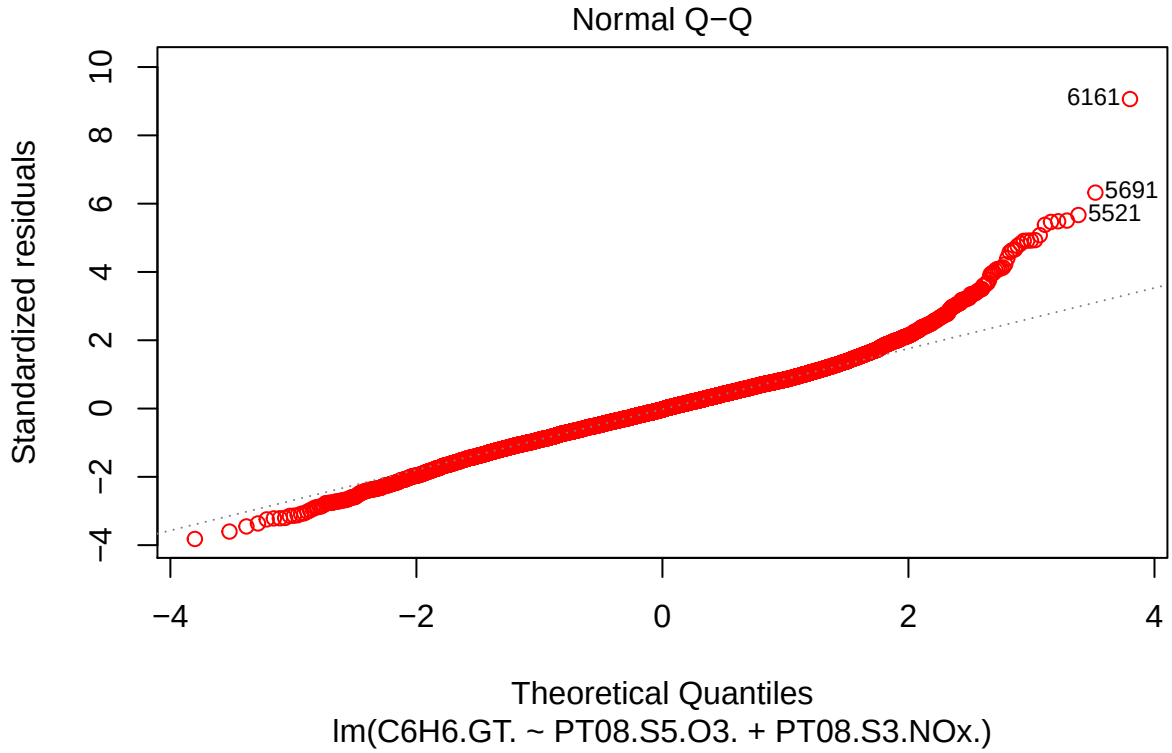
```
vif(fit3)
```

```
##  PT08.S5.O3. PT08.S3.NOx.
##     2.698521     2.698521
```

```r
plot(fit3, which=1, col=c("blue"))
```




```r
plot(fit3, which=2, col=c("red"))
```



We have some problems with residuals here.

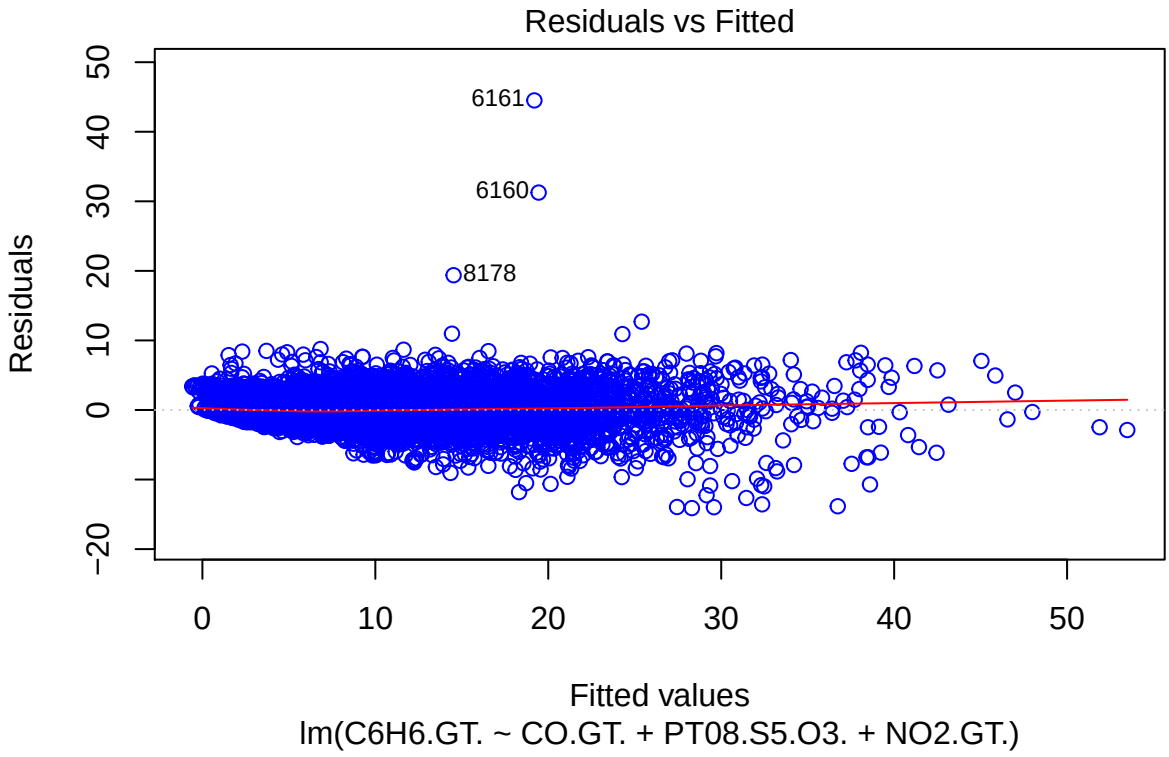
```r
fit4 <- lm(data = df_narm, formula =  C6H6.GT. ~ CO.GT. + PT08.S5.O3. + NO2.GT. )
summary(fit4)
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ CO.GT. + PT08.S5.O3. + NO2.GT., data = df_narm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.104  -1.358  -0.126   1.290  44.504
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.7403281  0.0986640  -17.64   <2e-16 ***
## CO.GT.       3.8804694  0.0407867   95.14   <2e-16 ***
## PT08.S5.O3.  0.0055342  0.0001501   36.86   <2e-16 ***
## NO2.GT.     -0.0178097  0.0009066  -19.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.5 on 6937 degrees of freedom
## Multiple R-squared:  0.8879, Adjusted R-squared:  0.8878
## F-statistic: 1.831e+04 on 3 and 6937 DF,  p-value: < 2.2e-16
```

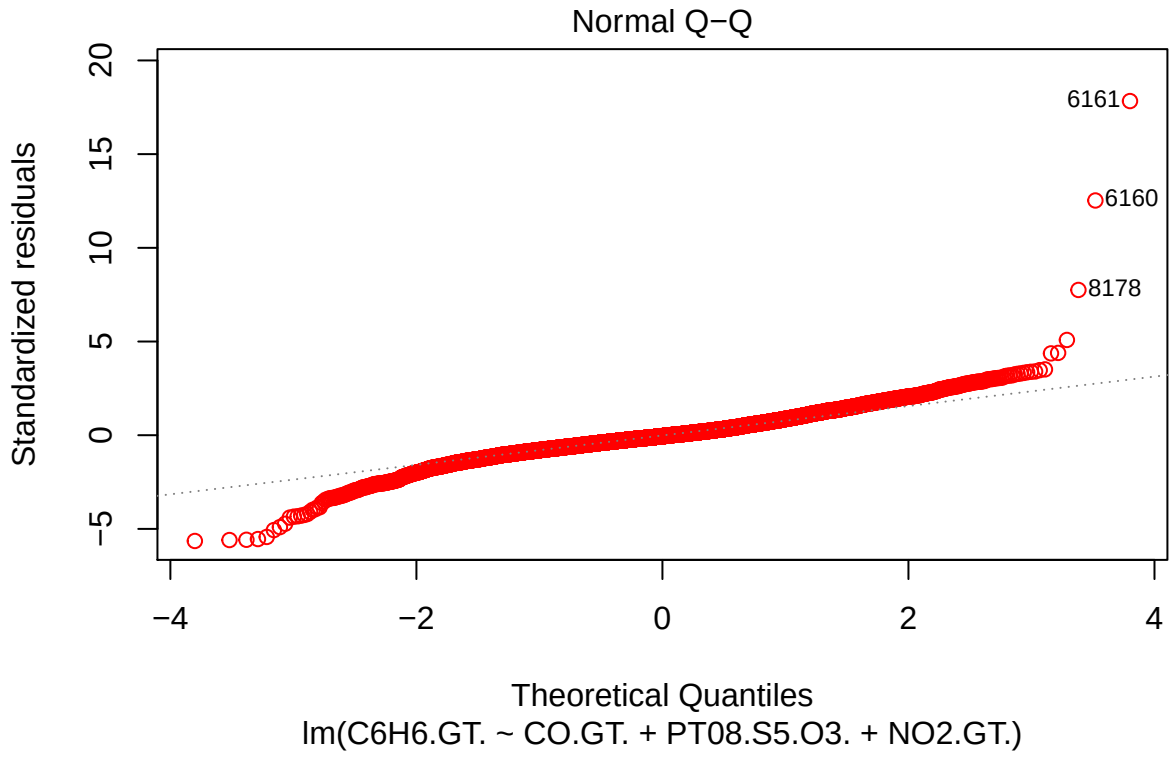
```r
vif(fit4)
```

```
##      CO.GT. PT08.S5.O3.     NO2.GT.
##    3.836025    4.135056    2.056937
```

```r
plot(fit4, which=1, col=c("blue"))
```



```r
plot(fit4, which=2, col=c("red"))
```



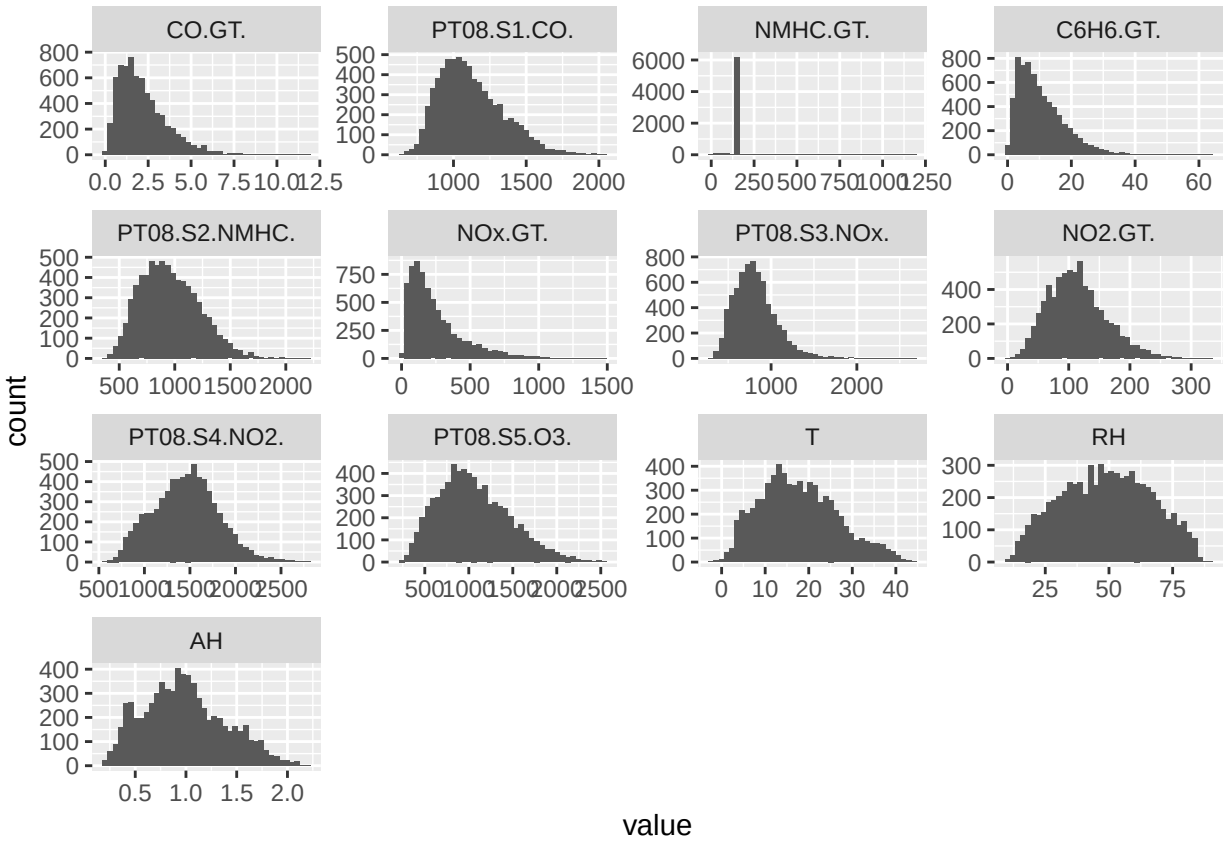
Residuals look ok, quantiles are ok, this model is the best one.

So, we will choose model 4: C6H6.GT. ~ CO.GT. + PT08.S5.O3. + NO2.GT.

Should we transform our data somehow? Let's check the distribution of variables.

```r
df_narm%>%
  mutate(id=c(1:nrow(df_narm)))%>%
  melt(measure.vars=c(1:13))->long

ggplot(long, aes(value)) +
  geom_histogram(bins=40) +
  facet_wrap(~variable, scales = "free")
```



Not all of the variables look normal. Let's try log transformation.

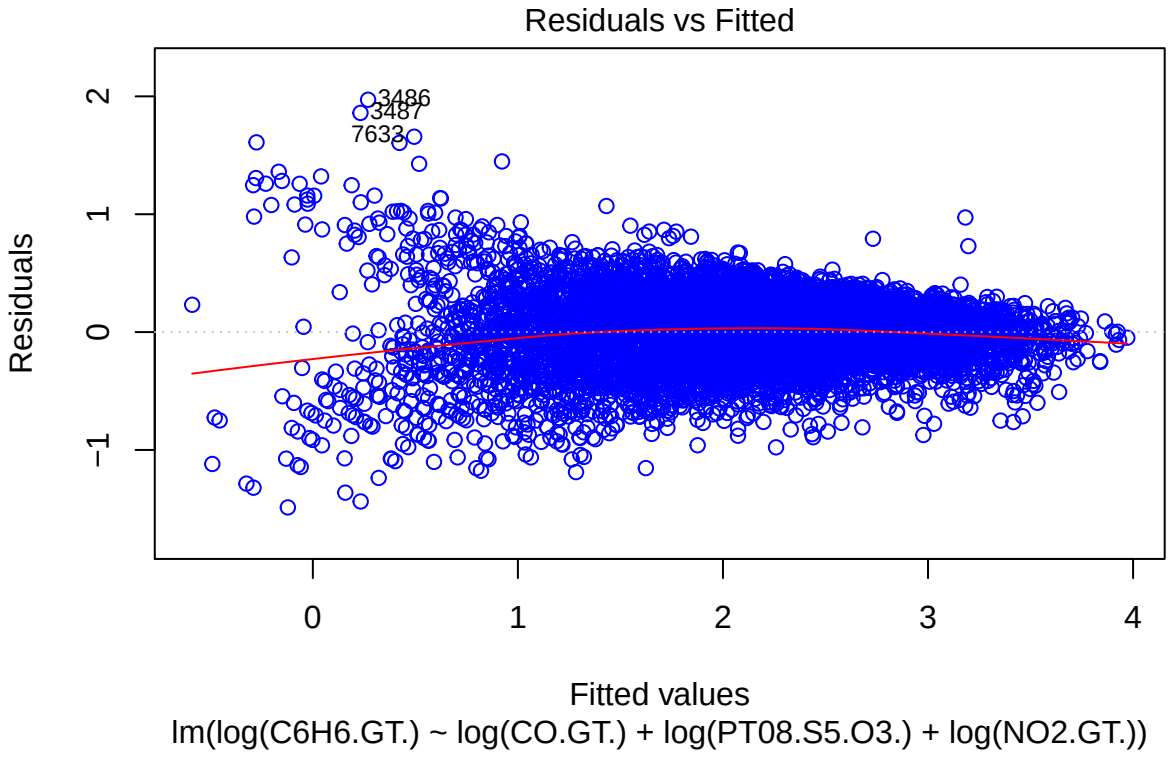
```
fitlog <- lm(data = df_narm, formula =  log(C6H6.GT.) ~ log(CO.GT.) + log(PT08.S5.O3.) + log(NO2.GT.) )
summary(fitlog)
```

```
##
## Call:
## lm(formula = log(C6H6.GT.) ~ log(CO.GT.) + log(PT08.S5.O3.) +
##     log(NO2.GT.), data = df_narm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48761 -0.18525  0.02355  0.18650  1.97121
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -4.431050   0.108738 -40.750  < 2e-16 ***
## log(CO.GT.)       0.578181   0.009656  59.875  < 2e-16 ***
## log(PT08.S5.O3.)  0.930869   0.016832  55.304  < 2e-16 ***
## log(NO2.GT.)     -0.046628   0.011911  -3.915 9.14e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3126 on 6937 degrees of freedom
## Multiple R-squared:  0.8491, Adjusted R-squared:  0.849
## F-statistic: 1.301e+04 on 3 and 6937 DF,  p-value: < 2.2e-16
```

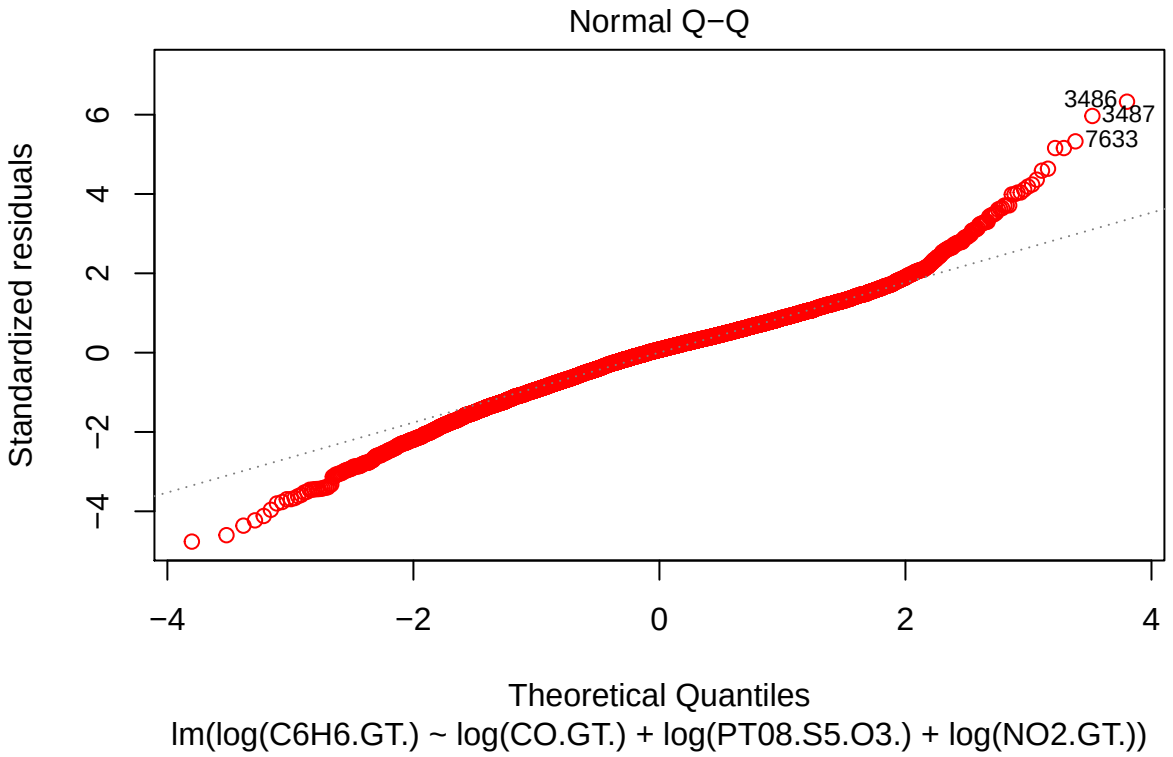
```
vif(fitlog)
```

```
##      log(CO.GT.) log(PT08.S5.O3.)     log(NO2.GT.)
##         3.432043         3.314865         2.218925
```

```
plot(fitlog, which=1, col=c("blue"))
```




```
plot(fitlog, which=2, col=c("red"))
```



Residuals look a bit skewed.

So, we choose model 4 as the final model (C6H6.GT. ~ CO.GT. + PT08.S5.O3. + NO2.GT.)