Loading libraries

##Air quality data set - Multiple linear regreession Data Set Information:

The dataset contains 9358 instances of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device. The device was located on the field in a significantly polluted area, at road level,within an Italian city. Data were recorded from March 2004 to February 2005 (one year)representing the longest freely available recordings of on field deployed air quality chemical sensor devices responses. Ground Truth hourly averaged concentrations for CO, Non Metanic Hydrocarbons, Benzene, Total Nitrogen Oxides (NOx) and Nitrogen Dioxide (NO2) and were provided by a co-located reference certified analyzer. Evidences of cross-sensitivities as well as both concept and sensor drifts are present as described in De Vito et al., Sens. And Act. B, Vol. 129,2,2008 (citation required) eventually affecting sensors concentration estimation capabilities. Missing values are tagged with -200 value. This dataset can be used exclusively for research purposes. Commercial purposes are fully excluded.

Attribute Information:

0 Date (DD/MM/YYYY) 1 Time (HH.MM.SS) 2 True hourly averaged concentration CO in mg/mˆ3 (reference analyzer) 3 PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted) 4 True hourly averaged overall Non Metanic HydroCarbons concentration in microg/mˆ3 (reference analyzer) 5 True hourly averaged Benzene concentration in microg/mˆ3 (reference analyzer) 6 PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted) 7 True hourly averaged NOx concentration in ppb (reference analyzer) 8 PT08.S3 (tungsten oxide) hourly averaged sensor response (nominally NOx targeted) 9 True hourly averaged NO2 concentration in microg/mˆ3 (reference analyzer) 10 PT08.S4 (tungsten oxide) hourly averaged sensor response (nominally NO2 targeted) 11 PT08.S5 (indium oxide) hourly averaged sensor response (nominally O3 targeted) 12 Temperature in Â°C 13 Relative Humidity (%) 14 AH Absolute Humidity

Loading csv file

```
AQ_raw <- read.csv("/home/aleksandar/Desktop/R statistics 2/hw_1_Rstat2/AirQualityUCI/AirQualityUCI.csv
```

Removing NAs Since missing values are denoted with -200 in this data set, first step would be to change all -200 values to NA and then remove NAs

```
AQ_NAs <- AQ_raw %>% na_if(-200) %>% na_if("-200,0")
#removing empty columns
toDrop <- c("X", "X.1")
AQ_NAs <- AQ_NAs[,!(names(AQ_NAs)) %in% toDrop]
```

Exploring NAs, after visually checking data set it seems most NAs are from NMHC.GT column

```
sapply(AQ_NAs, function(x) sum(is.na(x)))
```

```
##          Date          Time        CO.GT.   PT08.S1.CO.      NMHC.GT.
##             0             0          1683           480          8557
##      C6H6.GT. PT08.S2.NMHC.       NOx.GT.  PT08.S3.NOx.       NO2.GT.
##           366           480          1753           480          1756
##  PT08.S4.NO2.   PT08.S5.O3.             T            RH            AH
##           480           480           366           366           366
```

Since NMHC.GT has a lot of NAs using complete cases would drop about 90% of cases. I think its a good option to remove NMHC.GT from the main analyses.

```
toDrop <- c("NMHC.GT.", "Date", "Time") #also dropped Date and Time colums since I wont be using them
AQ_NAs <- AQ_NAs[,!(names(AQ_NAs)) %in% toDrop]
complete_AQrows <- complete.cases(AQ_NAs)
AQ <- AQ_NAs[complete_AQrows,]
```

Checking for non numeric variables

```
sapply(AQ, function(x)is.numeric(x))
```

```
##         CO.GT.    PT08.S1.CO.       C6H6.GT. PT08.S2.NMHC.        NOx.GT.
##          FALSE           TRUE          FALSE           TRUE           TRUE
##  PT08.S3.NOx.        NO2.GT.  PT08.S4.NO2.    PT08.S5.O3.              T
##           TRUE           TRUE           TRUE           TRUE          FALSE
##             RH             AH
##          FALSE          FALSE
```

```
AQnum <- data.frame(sapply(AQ, function(x) as.double(gsub(",",".",x)))) #converting to double

sapply(AQnum, function(x)is.numeric(x))
```

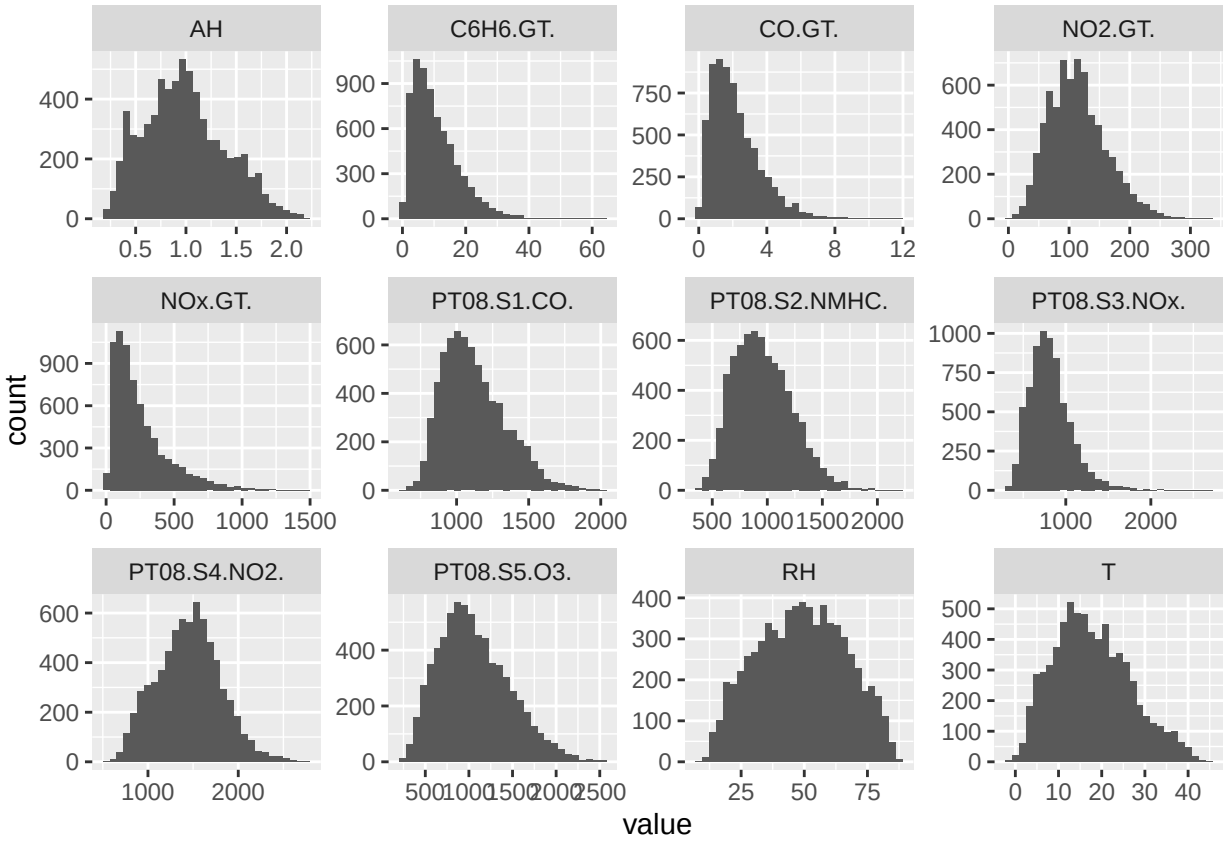
```
##         CO.GT.    PT08.S1.CO.       C6H6.GT. PT08.S2.NMHC.        NOx.GT.
##           TRUE           TRUE           TRUE           TRUE           TRUE
##  PT08.S3.NOx.        NO2.GT.  PT08.S4.NO2.    PT08.S5.O3.              T
##           TRUE           TRUE           TRUE           TRUE           TRUE
##             RH             AH
##           TRUE           TRUE
```

Exploring the data individually

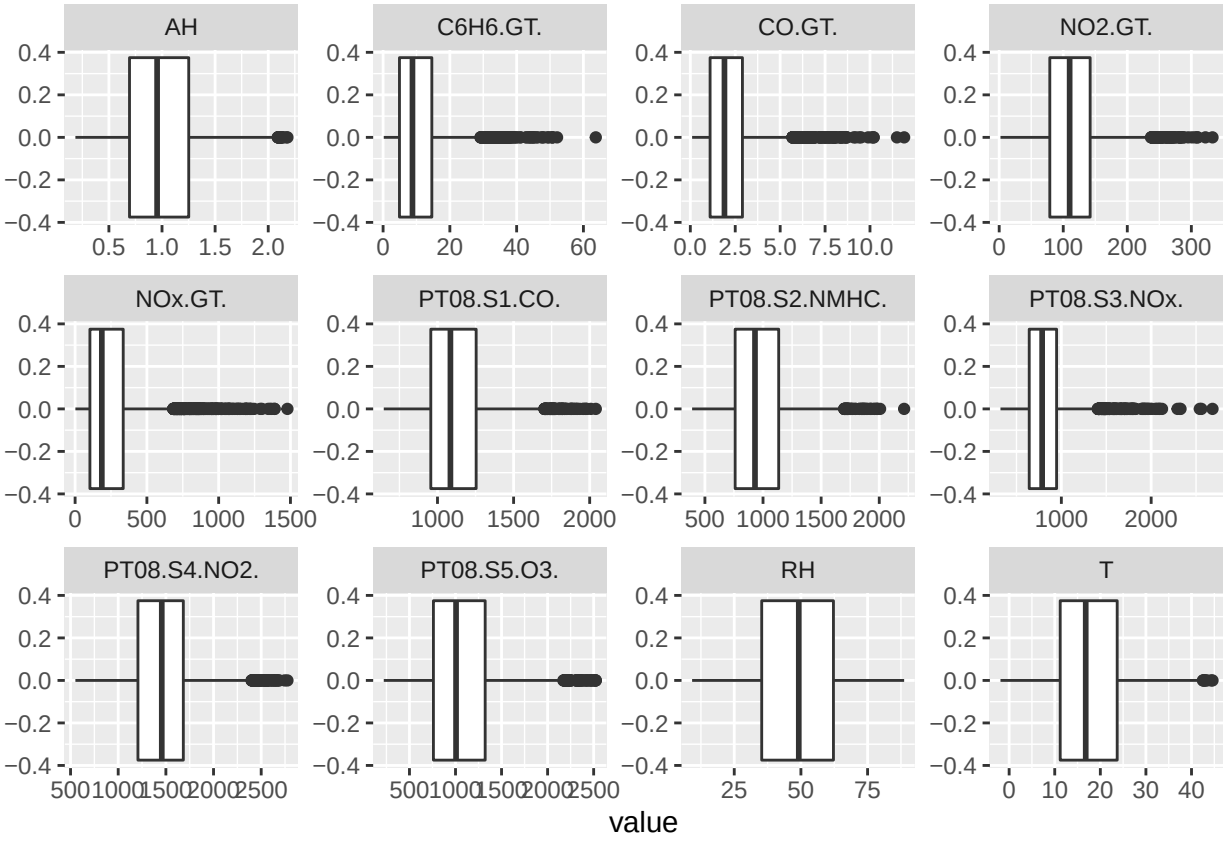
```
AQlong <- gather(AQnum)

ggplot(AQlong, aes(value)) + geom_histogram() + facet_wrap(~key, scales = "free")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



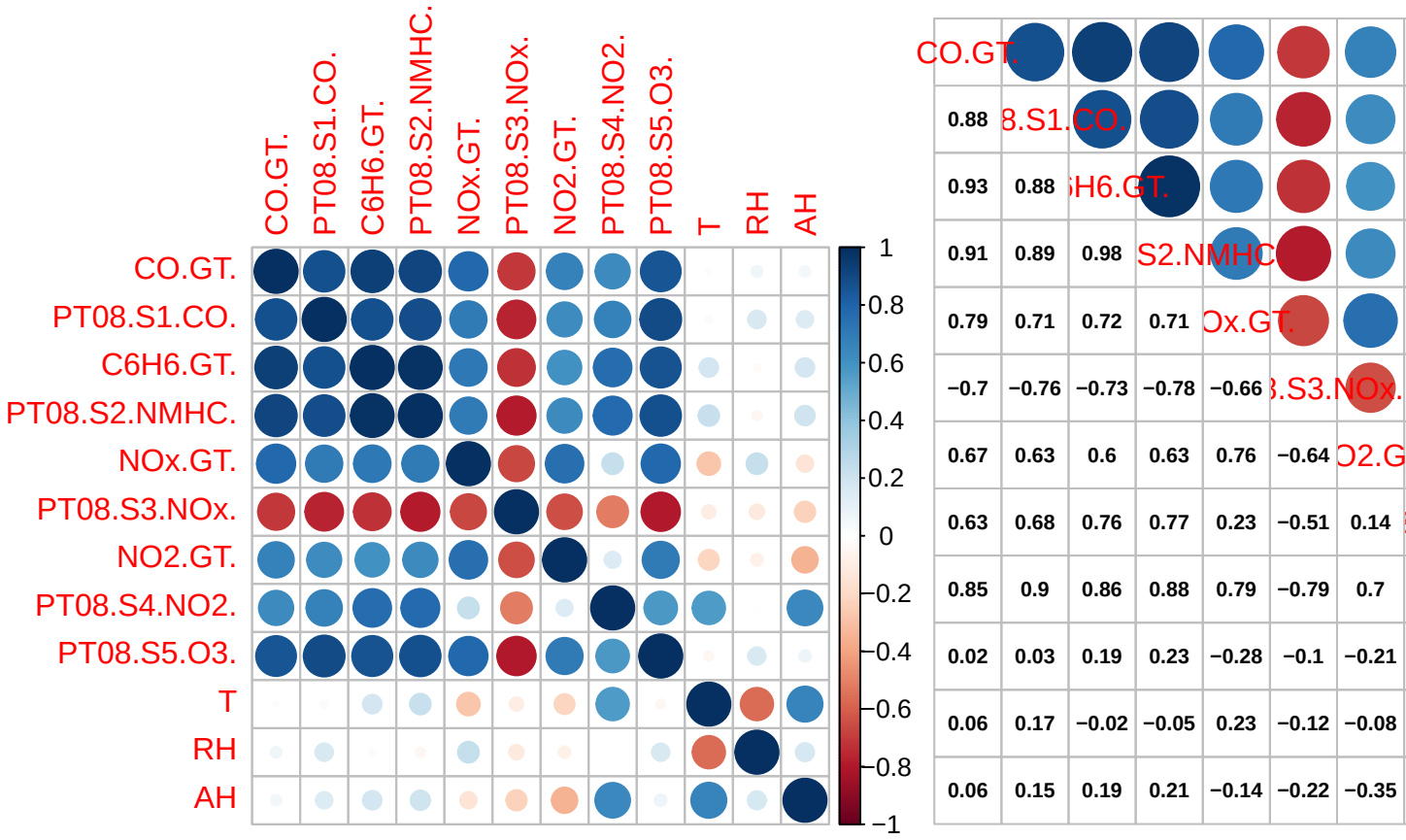
```
ggplot(AQlong, aes(value)) + geom_boxplot() + facet_wrap(~key, scales = "free")
```



Finding pairwise correlations for data set Response for further analyses is C6H6.GT.

Multicollinearity

```
corrplot.mixed(corrplot(cor(AQnum)), lower.col = "black", number.cex = .7)
```

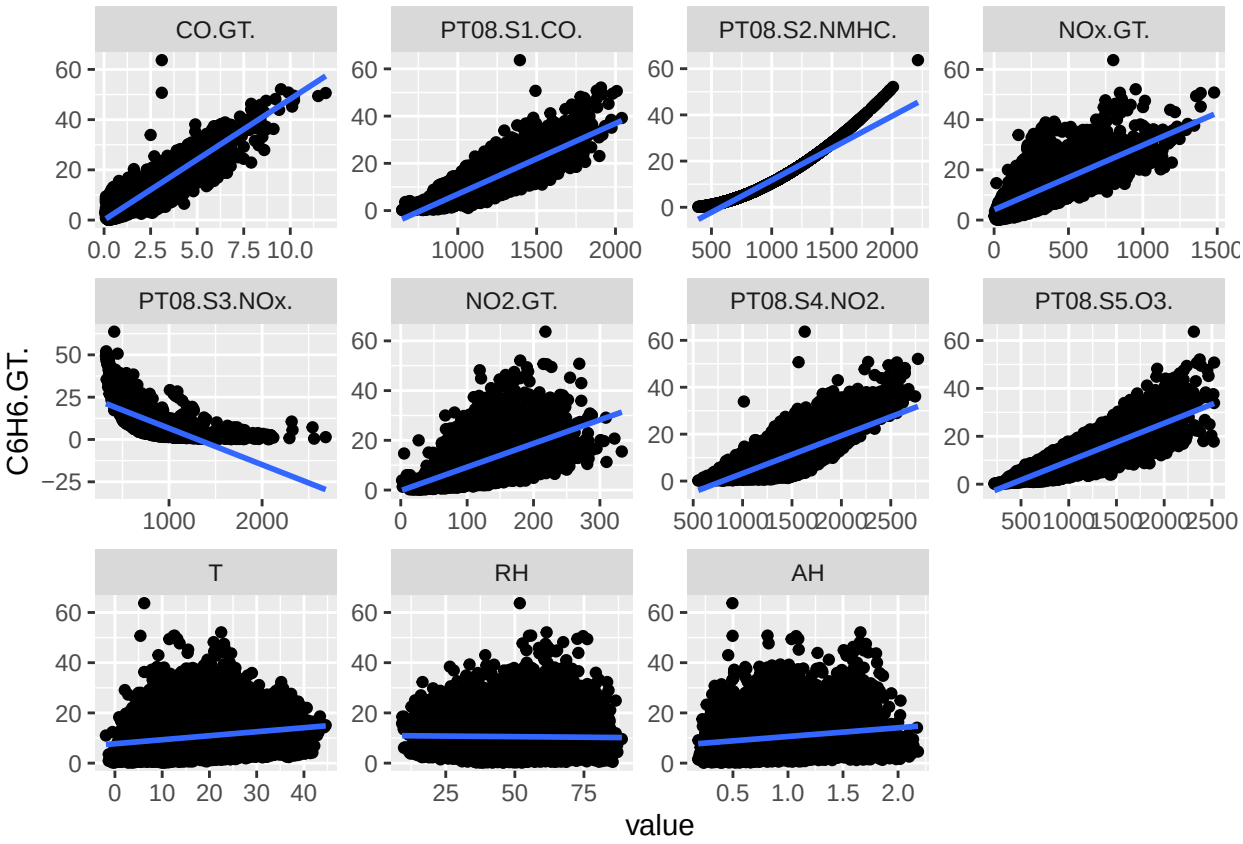


0.88	0.93	0.88	0.91	0.89	0.98	0.79	0.71	0.72	0.71	−0.7	−0.76	−0.73	−0.78	−0.66	0.67	0.63	0.6	0.63	0.76	−0.64	0.63	0.68	0.76	0.77	0.23	−0.51	0.14	0.85	0.9	0.86	0.88	0.79	−0.79	0.7	0.02	0.03	0.19	0.23	−0.28	−0.1	−0.21	0.06	0.17	−0.02	−0.05	0.23	−0.12	−0.08	0.06	0.15	0.19	0.21	−0.14	−0.22	−0.35

##Visualising and inspecting some pairs of columns more closely

```
d <- melt(AQnum, id.vars = "C6H6.GT.")
ggplot(d, aes(x = value, y = C6H6.GT.)) + geom_point() + geom_smooth(method = lm) + facet_wrap(~variable
```

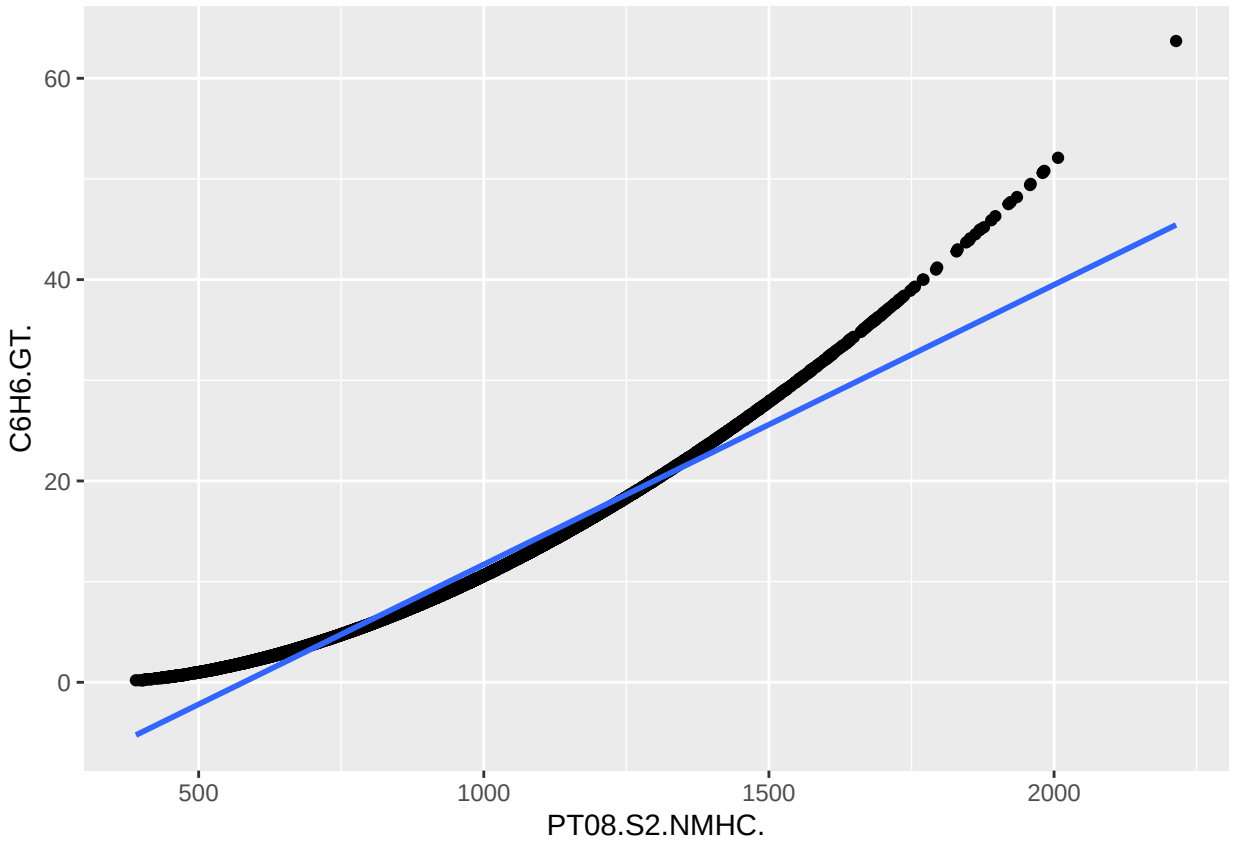
```
## `geom_smooth()` using formula 'y ~ x'
```



Before constructing a model I decided to look into NMHC vs C6H6, there seems to be very little devieation here, but I suspect that the curve of the scatter plot isnt suited for a linear model.

```
ggplot(AQnum, aes(x = PT08.S2.NMHC., y = C6H6.GT.)) + geom_point() + geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



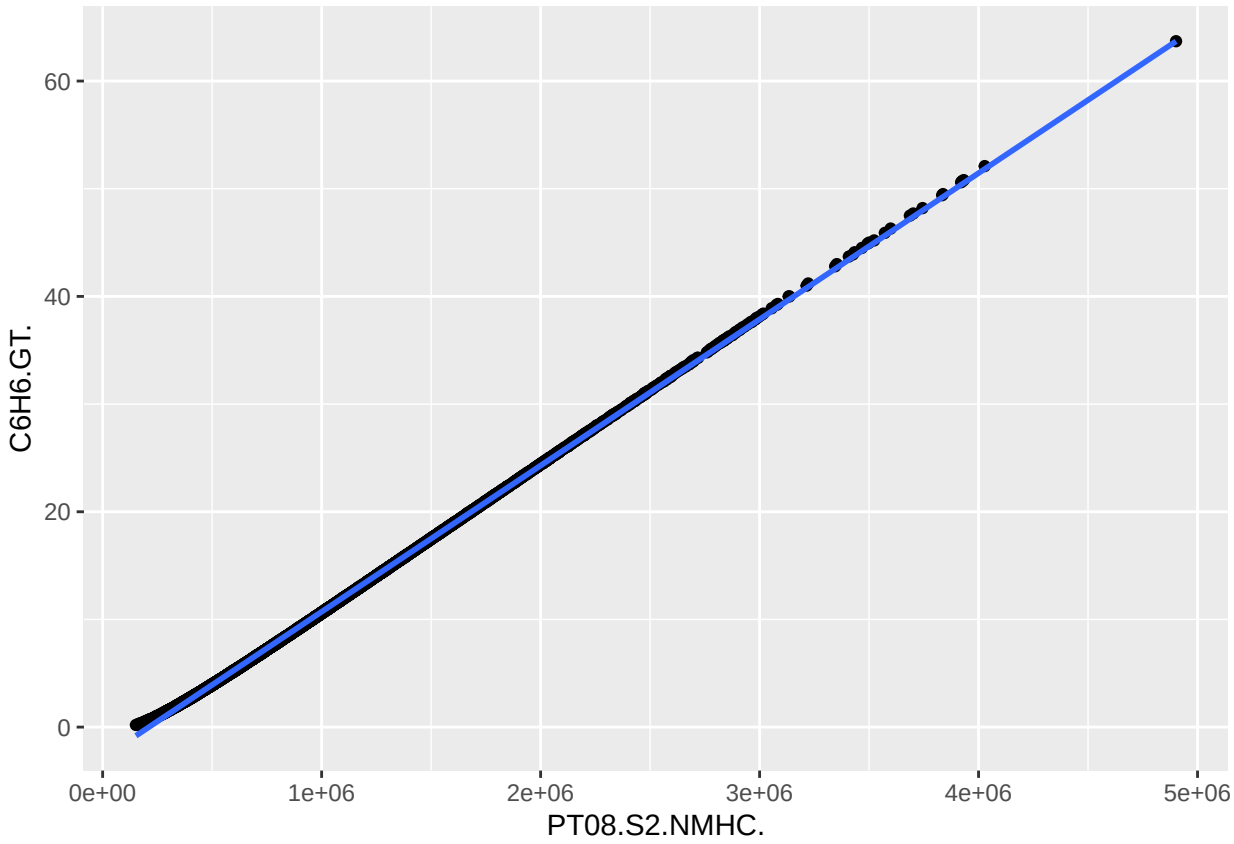
I tried squaring the NMCH column to se if it would improve result

```
AQnum$PT08.S2.NMHC. <- AQ$PT08.S2.NMHC.^2
```

Not it is almost a perfect fit

```
ggplot(AQnum, aes(x = PT08.S2.NMHC., y = C6H6.GT.)) + geom_point() + geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Multiple regression model - predictors: CO.GT. + PT08.S1.CO. + PT08.S2.NMHC. + PT08.S4.NO2. + PT08.S5.O3.

```
ML_mod <- lm(formula = C6H6.GT. ~ CO.GT. +  PT08.S1.CO. + PT08.S2.NMHC. + PT08.S4.NO2. + PT08.S5.O3. , d
summary(ML_mod)
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ CO.GT. + PT08.S1.CO. + PT08.S2.NMHC. +
##     PT08.S4.NO2. + PT08.S5.O3., data = AQnum)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36536 -0.08978 -0.02887  0.05909  0.86352
##
## Coefficients:
##                Estimate Std. Error  t value Pr(>|t|)
## (Intercept)   -2.608e+00  1.461e-02 -178.517  < 2e-16 ***
## CO.GT.         2.115e-02  3.474e-03    6.089 1.19e-09 ***
## PT08.S1.CO.   -3.447e-05  2.038e-05   -1.691   0.0908 .
## PT08.S2.NMHC.  1.375e-05  1.112e-08 1236.279  < 2e-16 ***
## PT08.S4.NO2.  -1.436e-04  8.022e-06  -17.900  < 2e-16 ***
## PT08.S5.O3.   -2.130e-04  1.021e-05  -20.864  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1347 on 6935 degrees of freedom
## Multiple R-squared:  0.9997, Adjusted R-squared:  0.9997
```

```
## F-statistic: 4.264e+06 on 5 and 6935 DF,  p-value: < 2.2e-16
```

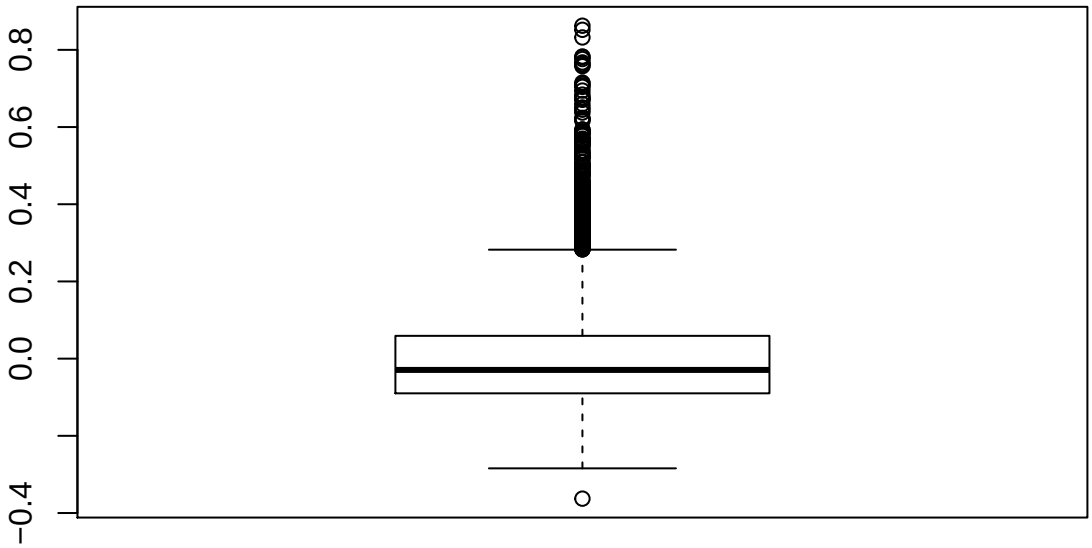
PT08.S1.CO. has lowered pvalue when included in multiple regression model in contrast when being the sole predictor. I will remove it from the model since $0.0908 > 0.05$

```
ML_mod <- lm(formula = C6H6.GT. ~ CO.GT. + PT08.S2.NMHC. + PT08.S4.NO2. + PT08.S5.O3. , data = AQnum)
summary(ML_mod)
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ CO.GT. + PT08.S2.NMHC. + PT08.S4.NO2. +
##     PT08.S5.O3., data = AQnum)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36273 -0.08994 -0.02925  0.05902  0.86249
##
## Coefficients:
##                 Estimate Std. Error  t value Pr(>|t|)
## (Intercept)   -2.627e+00  9.341e-03 -281.202  < 2e-16 ***
## CO.GT.         1.932e-02  3.302e-03    5.853 5.05e-09 ***
## PT08.S2.NMHC.  1.375e-05  1.111e-08 1237.045  < 2e-16 ***
## PT08.S4.NO2.  -1.476e-04  7.668e-06  -19.246  < 2e-16 ***
## PT08.S5.O3.   -2.230e-04  8.350e-06  -26.701  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1347 on 6936 degrees of freedom
## Multiple R-squared:  0.9997, Adjusted R-squared:  0.9997
## F-statistic: 5.329e+06 on 4 and 6936 DF,  p-value: < 2.2e-16
```

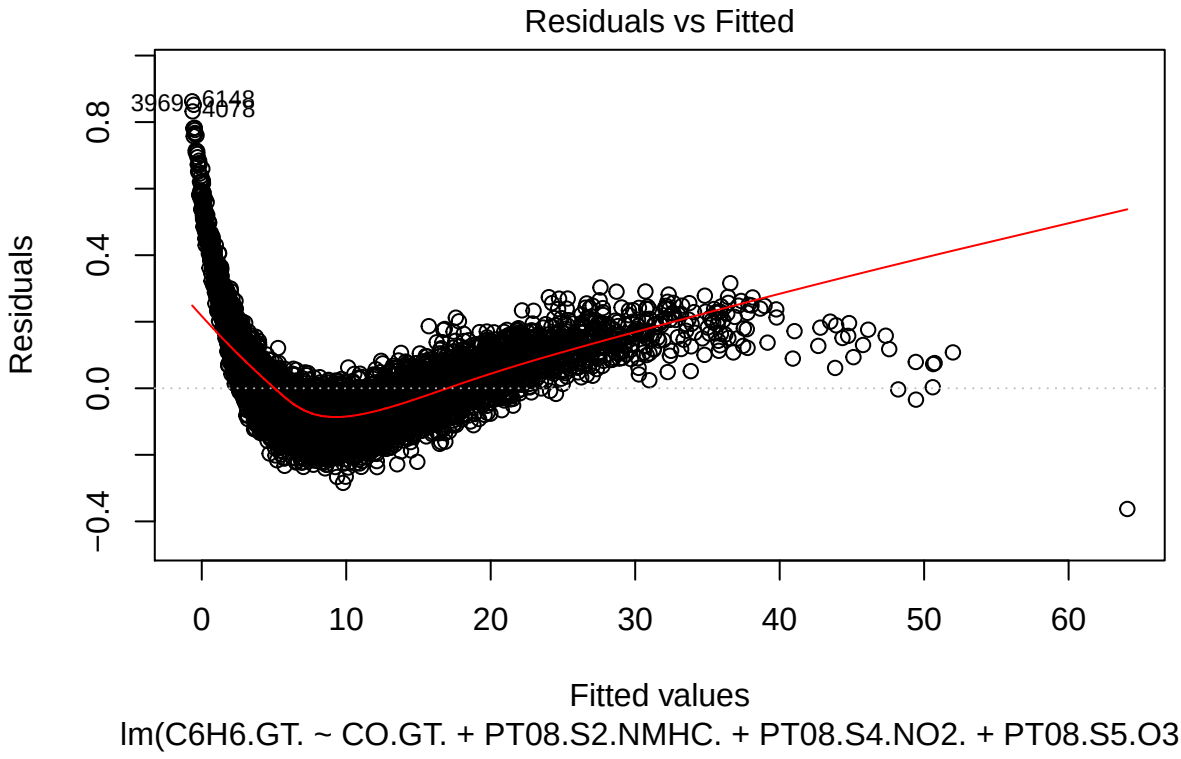
This doesnt look so good, residuals are not symetric

```
boxplot(ML_mod$residuals)
```



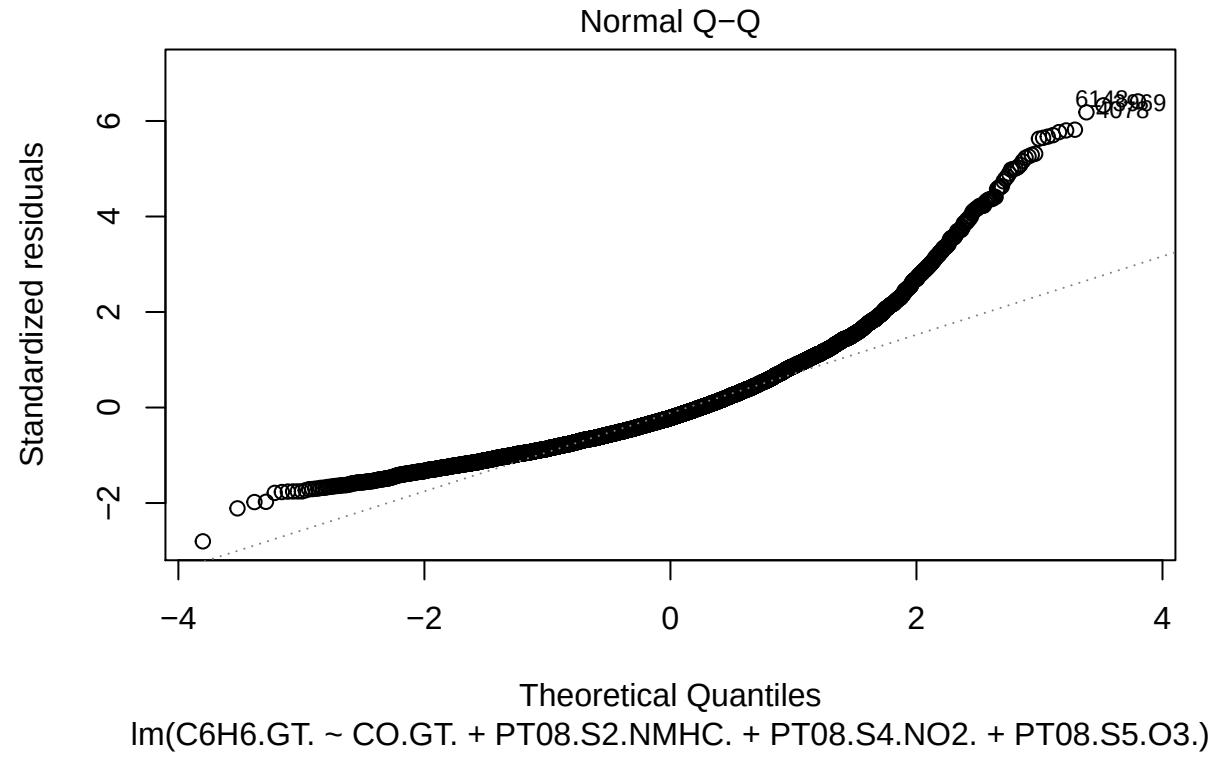
Residuals vs fitted - how well do fitted value align to residuals - it means residuals are uneuqally spread around the data

```
plot(ML_mod, which = 1)
```



QQplot - are theoretical quantiles the same as actual ones - this means that residuals at one point are not random and that there is something skewing the model

```
plot(ML_mod, which = 2)
```



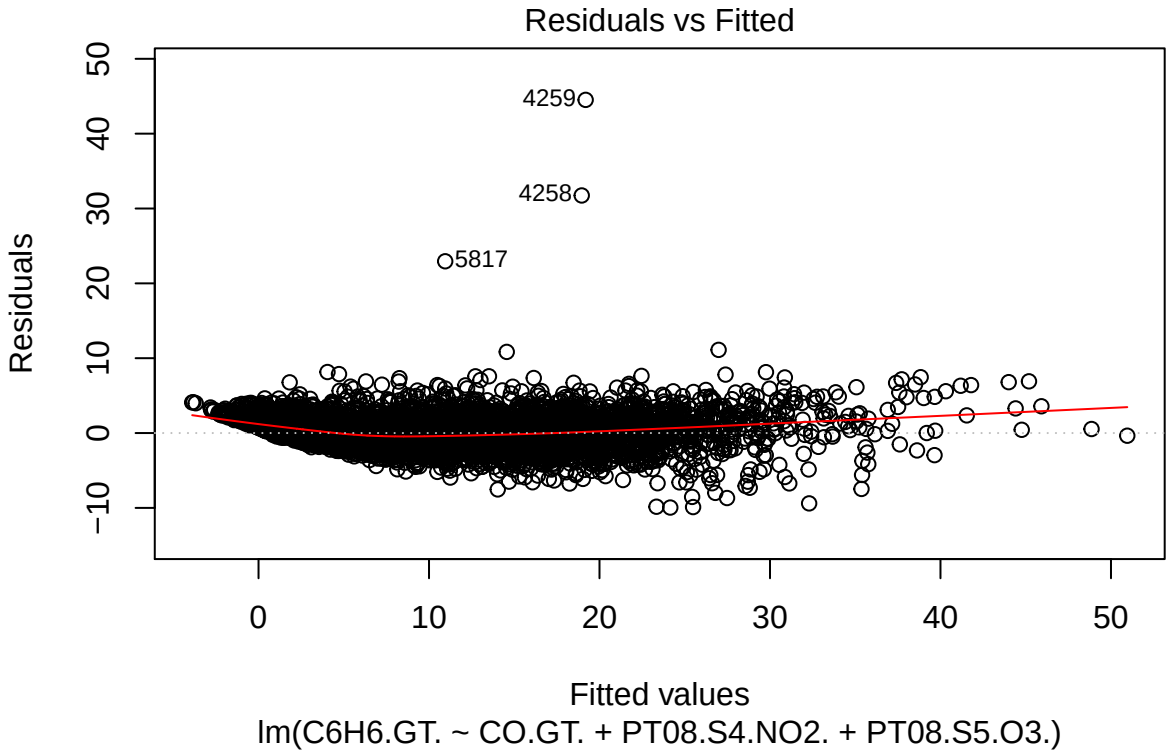
Try to build model again and repeat analysis, removing NMHC from model

```r
ML_mod <- lm(formula = C6H6.GT. ~ CO.GT. + PT08.S4.NO2. + PT08.S5.O3. , data = AQnum)
summary(ML_mod)
```

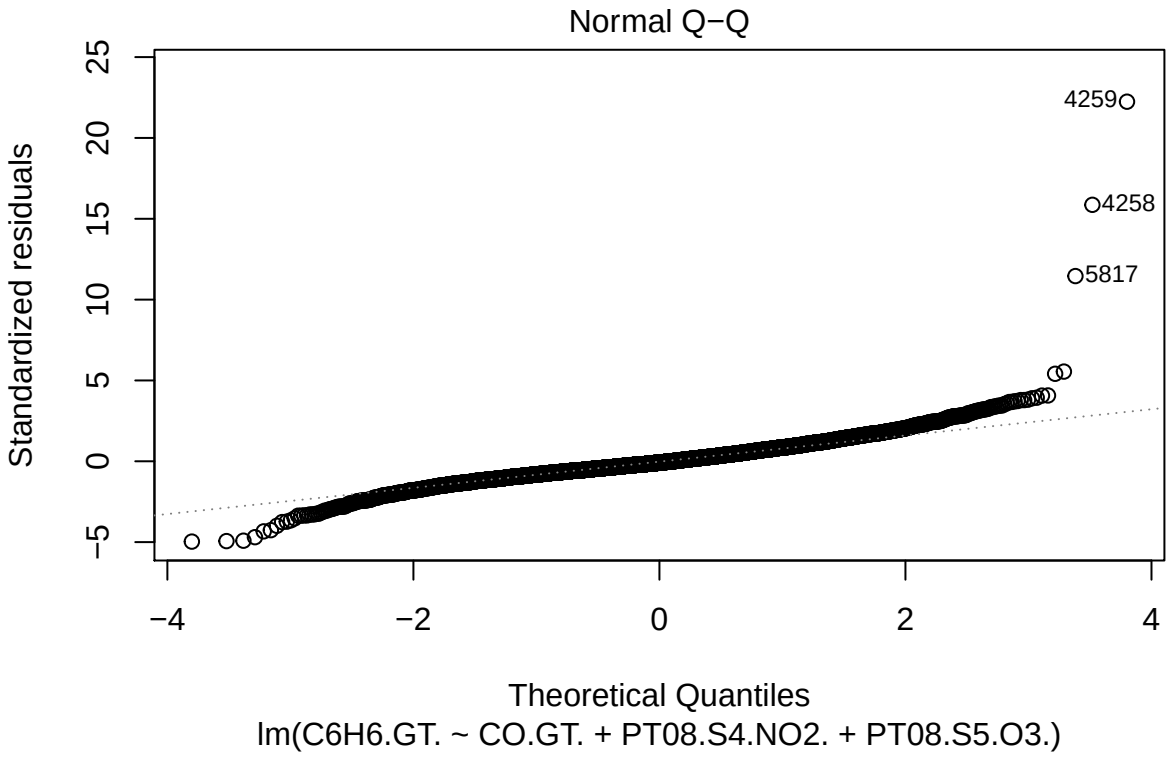
```
##
## Call:
## lm(formula = C6H6.GT. ~ CO.GT. + PT08.S4.NO2. + PT08.S5.O3.,
##     data = AQnum)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.952 -1.142 -0.169  1.050 44.513
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.584e+00  1.191e-01  -72.05   <2e-16 ***
## CO.GT.        2.973e+00  3.394e-02   87.59   <2e-16 ***
## PT08.S4.NO2.  5.880e-03  8.814e-05   66.72   <2e-16 ***
## PT08.S5.O3.   3.884e-03  1.140e-04   34.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.005 on 6937 degrees of freedom
## Multiple R-squared:  0.9279, Adjusted R-squared:  0.9279
## F-statistic: 2.976e+04 on 3 and 6937 DF,  p-value: < 2.2e-16
```

Now it looks acceptable

```r
plot(ML_mod, which = 1)
```




```r
plot(ML_mod, which = 2)
```



Perfoming train vs test data LM analysis

```
set.seed(42)
sample <- sample.int(n = nrow(AQnum), size = floor(0.75 * nrow(AQnum)))
train <- AQnum[sample,]
test <- AQnum[-sample,]

new_modCO <-  lm(formula = C6H6.GT. ~ CO.GT. + PT08.S4.NO2. + PT08.S5.O3. , data = train)
summary(new_modCO)
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ CO.GT. + PT08.S4.NO2. + PT08.S5.O3.,
##     data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.836 -1.150 -0.156  1.048 31.751
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.525e+00  1.330e-01  -64.09   <2e-16 ***
## CO.GT.        2.933e+00  3.837e-02   76.44   <2e-16 ***
## PT08.S4.NO2.  5.871e-03  9.893e-05   59.35   <2e-16 ***
## PT08.S5.O3.   3.914e-03  1.287e-04   30.40   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.958 on 5201 degrees of freedom
## Multiple R-squared:  0.9302, Adjusted R-squared:  0.9302
```

```
## F-statistic: 2.31e+04 on 3 and 5201 DF,  p-value: < 2.2e-16
```

Prediction using trained model

```
pred <- predict(new_modCO, newdata = test)
test$CO.GT._predicted <- pred
```

Constructing plot

```
ggplot() + geom_point(data = train, aes(x= CO.GT., y = C6H6.GT.)) + geom_smooth(data =train, aes(x=CO.G
```

```
## `geom_smooth()` using formula 'y ~ x'
```