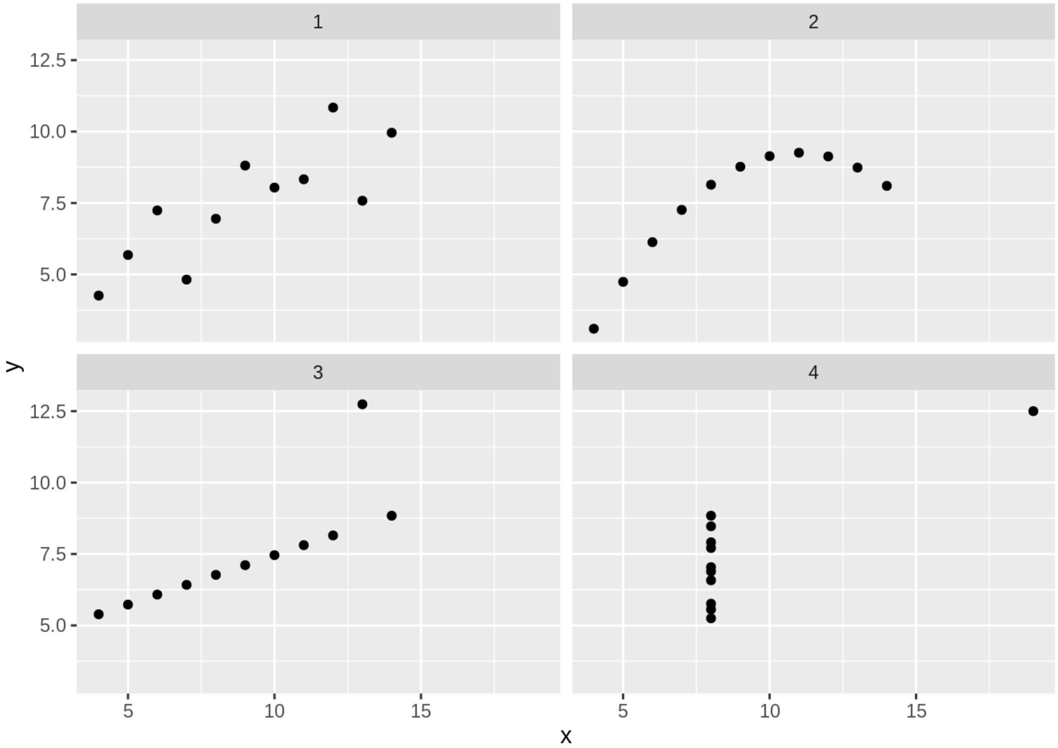
Task 1

Anscombe's data set

- Scatter plot facetted by set

```
data <- data.frame('x' = c(anscombe$x1, anscombe$x2, anscombe$x3, anscombe$x4),
                   'y' = c(anscombe$y1, anscombe$y2, anscombe$y3, anscombe$y4),
                   'set' =  rep(c(1:4), each = 11))
ggplot(data, aes(x, y)) +
  geom_point() +
  facet_wrap(~ set)
```



- Summary calculation (mean, sd) grouped by set

```
setNames(aggregate(data[, 1:2], list(data$set), mean), c('set', 'mean_x', 'mean_y'))
```

```
##   set mean_x   mean_y
## 1   1      9 7,500909
## 2   2      9 7,500909
## 3   3      9 7,500000
## 4   4      9 7,500909
```

```
setNames(aggregate(data[, 1:2], list('Sd' = data$set), sd),  c('set', 'sd_x', 'sd_y'))
```

```
##   set     sd_x     sd_y
## 1   1 3,316625 2,031568
## 2   2 3,316625 2,031657
## 3   3 3,316625 2,030424
## 4   4 3,316625 2,030579
```

- Pearson's correlation by set, and non-parametric, and p-value

```
data %>% group_by(set) %>% summarise(cor_pearson = cor.test(x,y, method = 'pearson')$estimate,
                                     cor_kendall = cor.test(x,y, method = 'kendall')$estimate,
                                     cor_spearmen = cor.test(x,y, method = 'spearman')$estimate)
```

```
## # A tibble: 4 x 4
##     set cor_pearson cor_kendall cor_spearmen
##   <int>       <dbl>       <dbl>        <dbl>
## 1     1       0.816       0.636        0.818
## 2     2       0.816       0.564        0.691
## 3     3       0.816       0.964        0.991
## 4     4       0.817       0.426        0.5
```

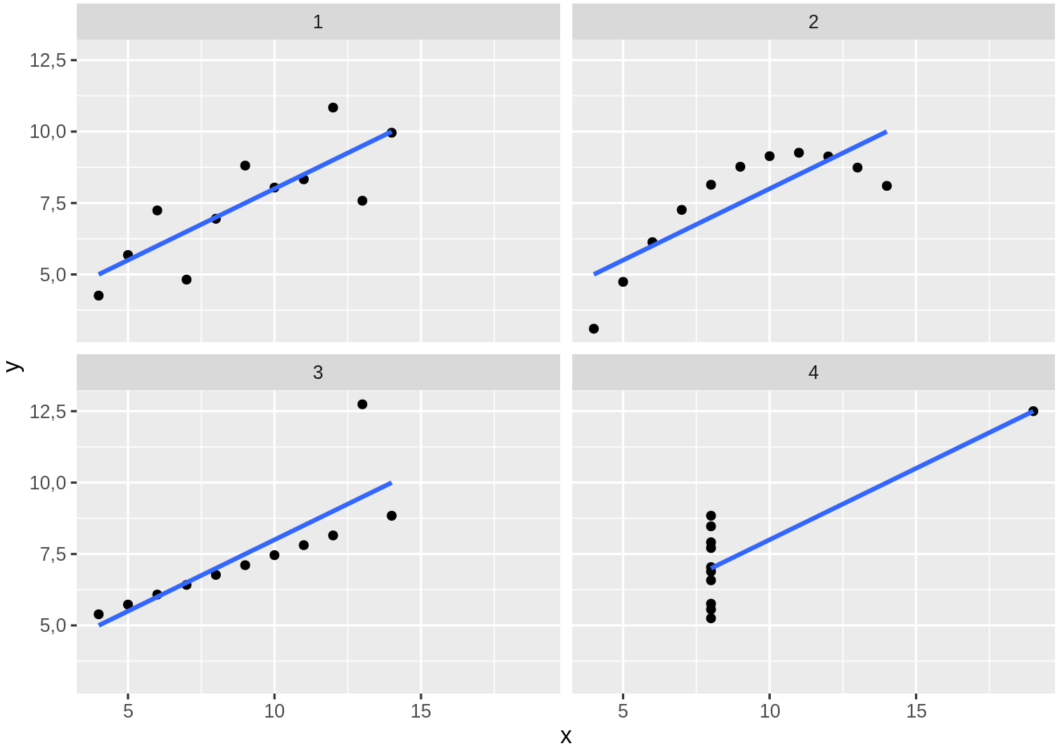
```
data %>% group_by(set) %>% summarise(p_pearson = cor.test(x,y, method = 'pearson')$p.value,
                                      p_kendall = cor.test(x,y, method = 'kendall')$p.value,
                                      p_spearmen = cor.test(x,y, method = 'spearman')$p.value)
```

```
## # A tibble: 4 x 4
##     set p_pearson   p_kendall p_spearmen
##   <int>     <dbl>       <dbl>      <dbl>
## 1     1   0.00217 0.00571        0.00373
## 2     2   0.00218 0.0165         0.0231
## 3     3   0.00218 0.000000551    0
## 4     4   0.00216 0.114          0.117
```

- Add geom_smooth() to the plot

```
ggplot(data, aes(x, y)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE) +
  facet_wrap(~ set)
```

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



Air Quality data set

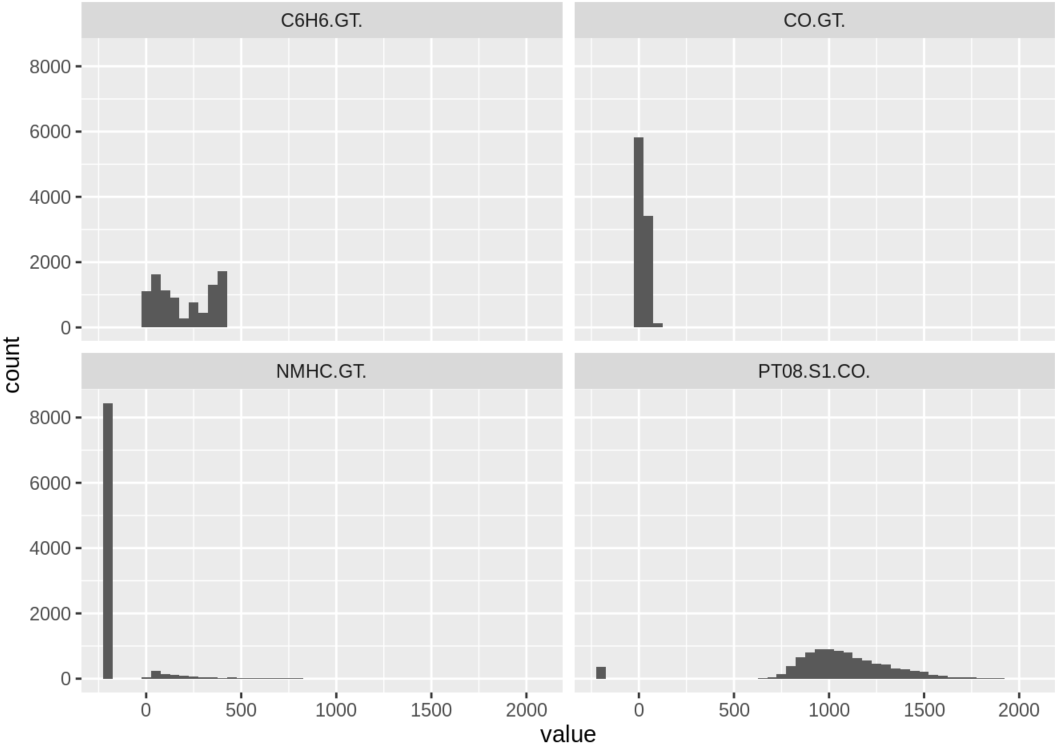
```
airq_data <- read.csv('/home/marina/Загрузки/AirQualityUCI.csv', sep = ';')
```

- Deleting NA columns and rows. Averaged concentration CO, Averaged Benzene concentration, Temperature, Humidity are factor columns in data, so it will be better to convert them into numeric. Actually, I don't need date and time for futher analysis, so I will slice the data.

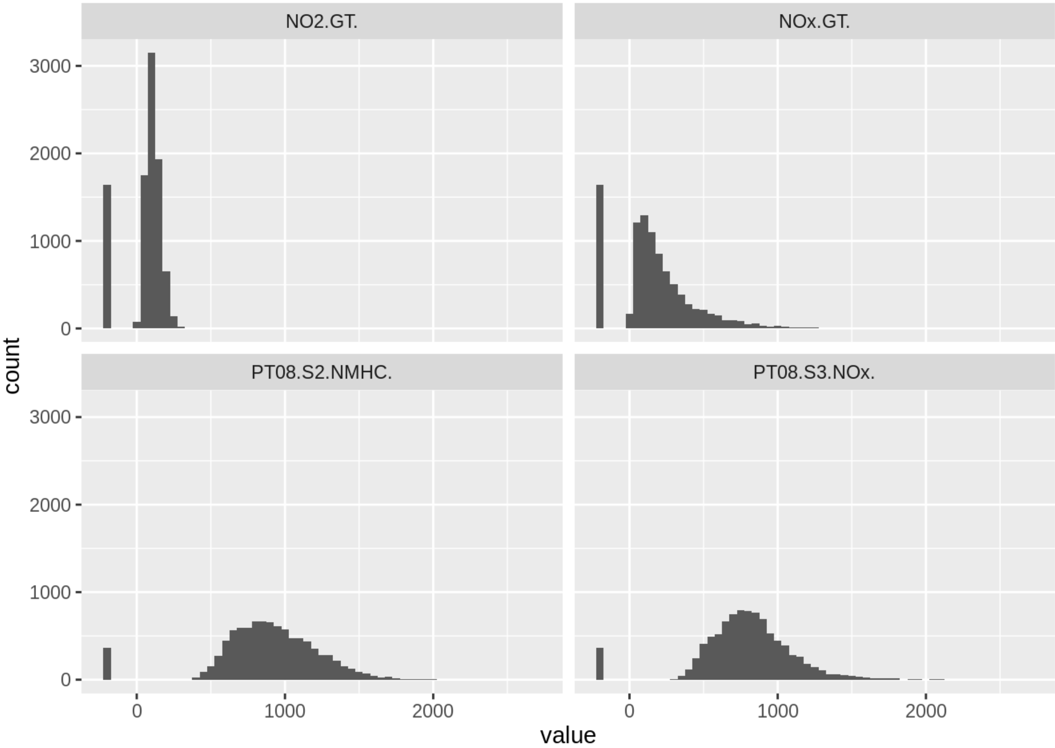
```
airq_data <- airq_data %>% select_if(~sum(!is.na(.)) > 0)  %>% drop_na()
airq_data[c('CO.GT.', 'C6H6.GT.', 'T', 'RH', 'AH')] <-  sapply(airq_data[c('CO.GT.', 'C6H6.GT.', 'T', 'RH', 'AH')], as.numeric)
airq_data <- airq_data[, c(3:15)]
```

- Exploring the variables (I make it in a 3 parts to better visualizing)

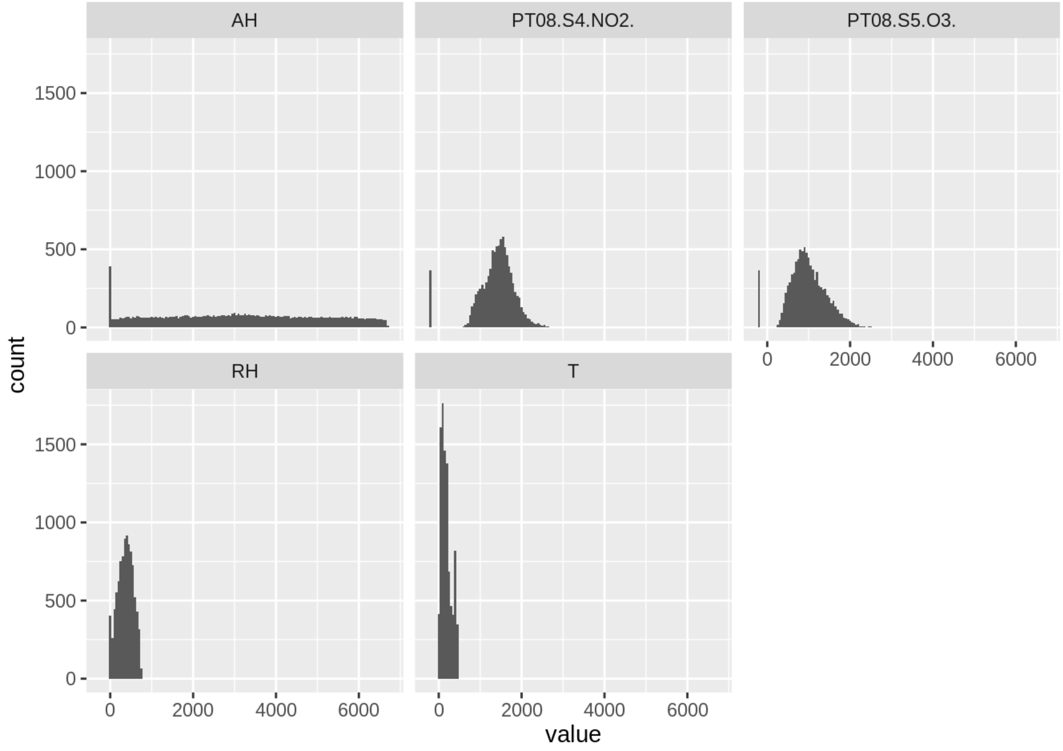
```
ggplot(gather(airq_data[, c(1:4)], cols, value), aes(x = value)) +
  geom_histogram(binwidth = 50) + facet_wrap(.~cols)
```




```r
ggplot(gather(airq_data[, c(5:8)], cols, value), aes(x = value)) +
  geom_histogram(binwidth = 50) + facet_wrap(.~cols)
```




```r
ggplot(gather(airq_data[, c(9:13)], cols, value), aes(x = value)) +
  geom_histogram(binwidth = 50) + facet_wrap(.~cols)
```



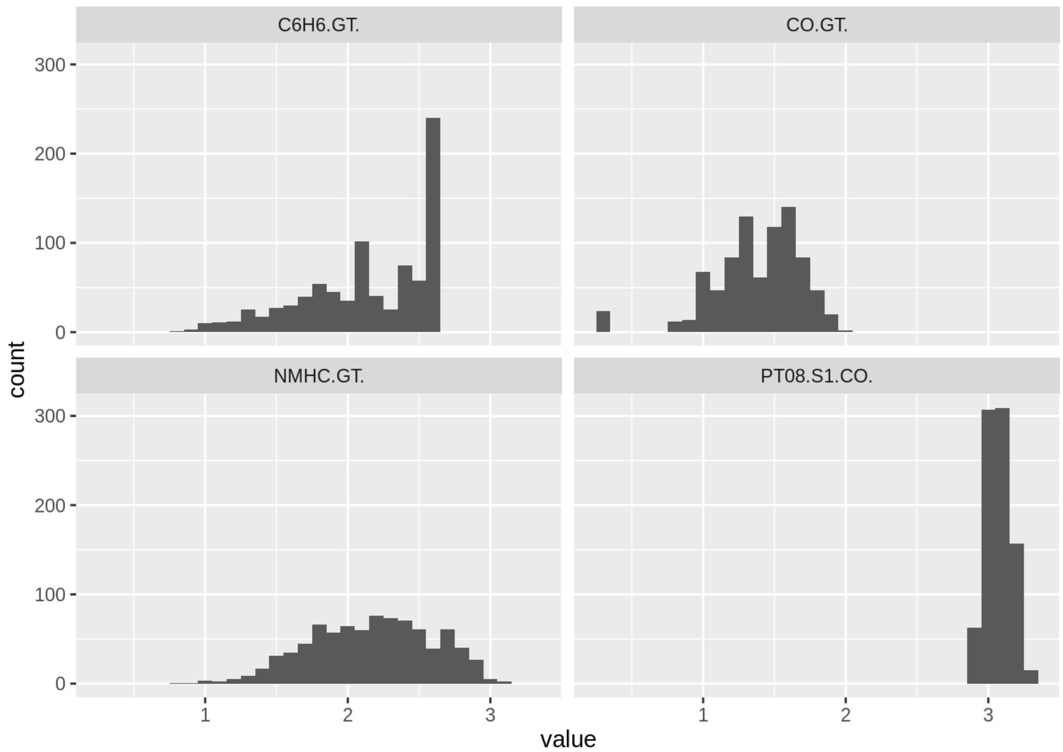
Ok, so it looks like a lot of variables have outliers and all of them in a different scale range So, as for me, it will be better to log-transform the data

- Log-transformation

```
airq_data <- log10(airq_data) %>% drop_na()
```

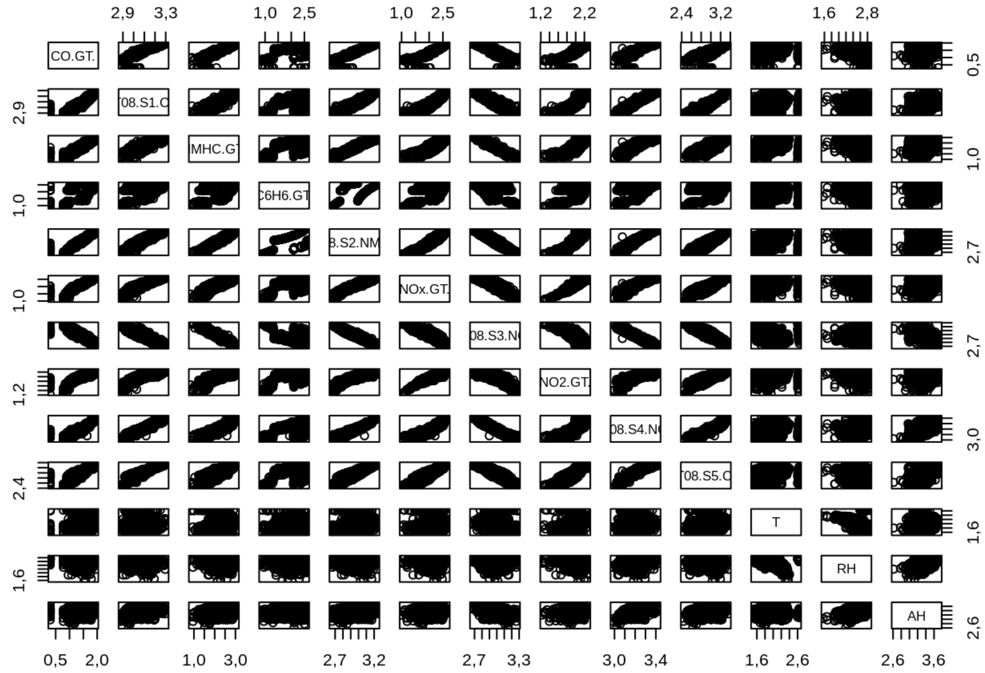
So, let's look now as an example for the first 4 columns:

```
ggplot(gather(airq_data[, c(1:4)], cols, value), aes(x = value)) +
  geom_histogram(binwidth = 0.1) + facet_wrap(.~cols)
```

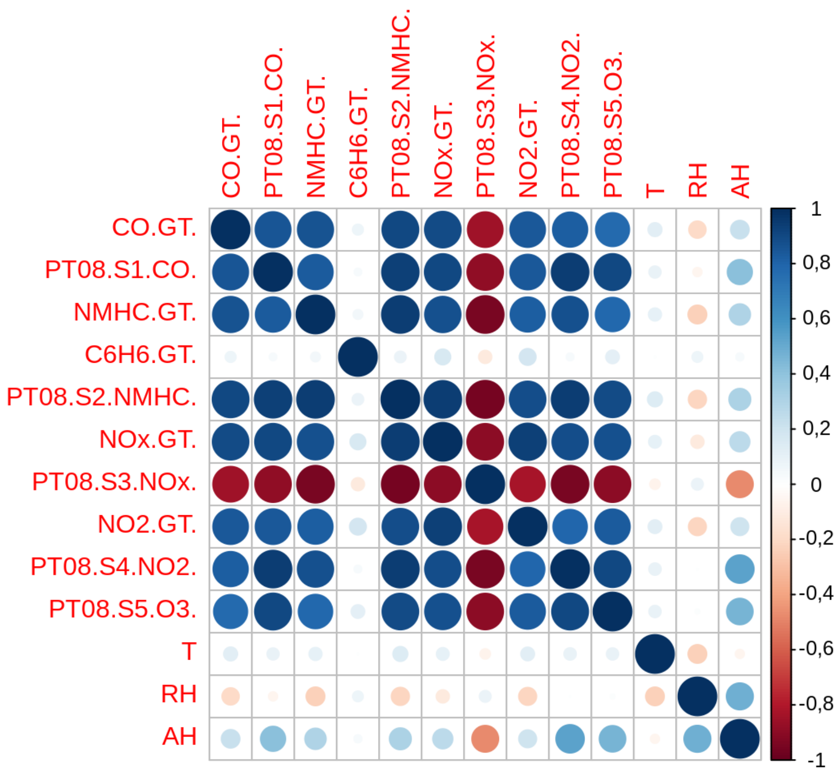


- Relationships between all variables

```
pairs(airq_data)
```




```
cor_data <- cor(airq_data)
corrplot(cor_data, method = 'circle')
```

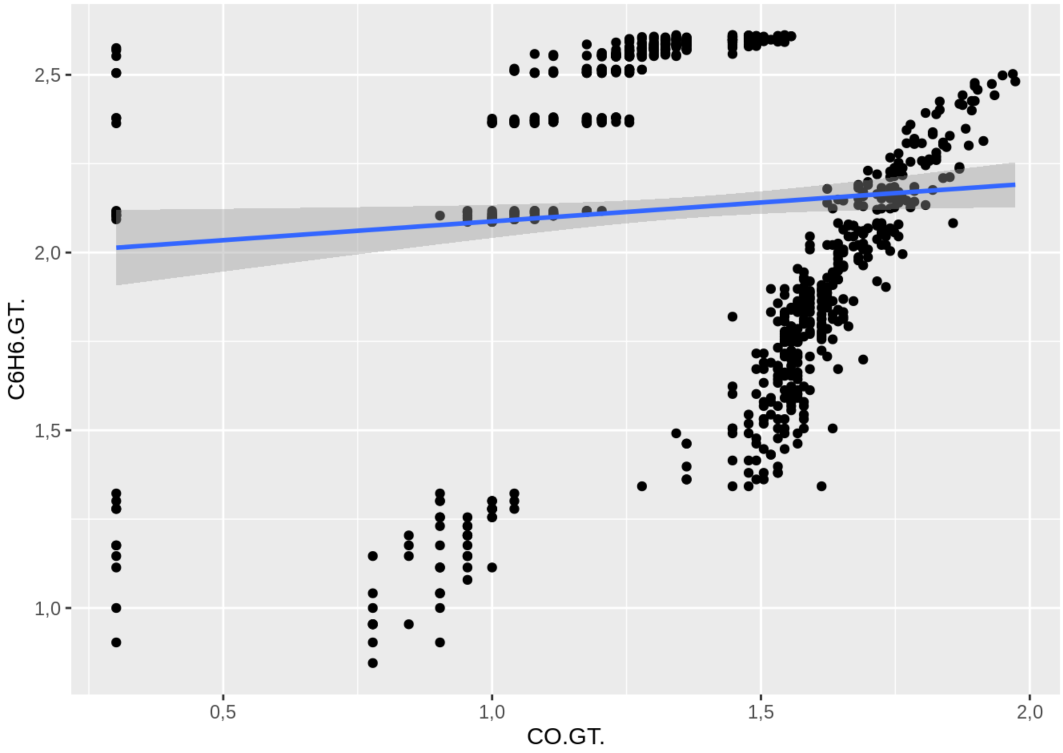


We can see that C6H6.GT. doesn't have a nice correlation with other variables. An if we check the assumptions for each variable we will see, that mostly all of them are not adhered.

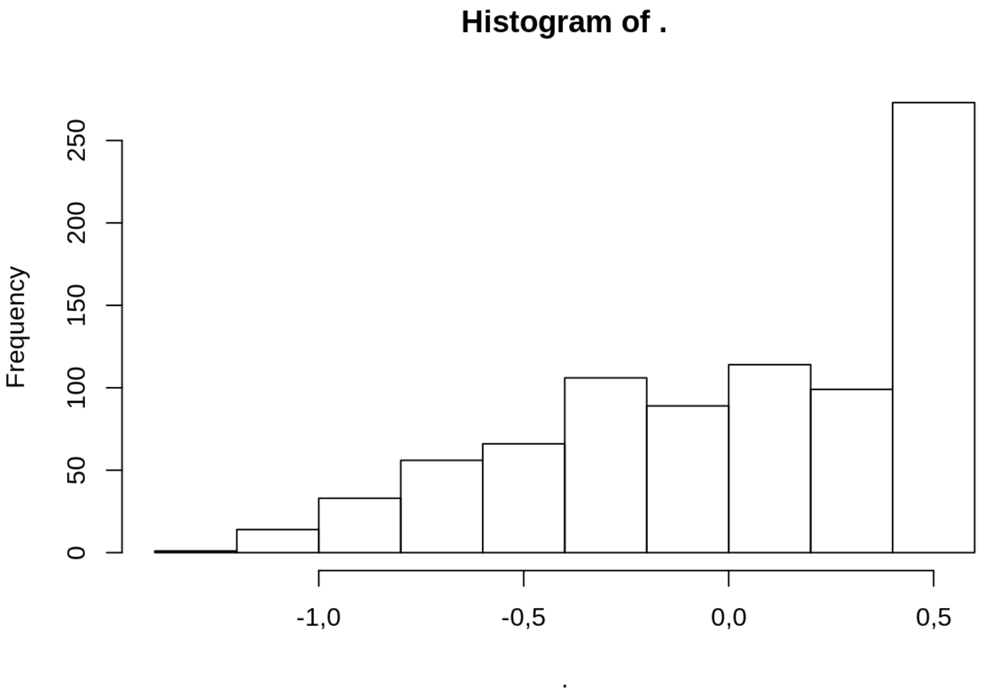
Example:

```
ggplot(airq_data, aes(x = CO.GT., y = C6H6.GT. )) +
  geom_point() +
  geom_smooth(method = 'lm')
```

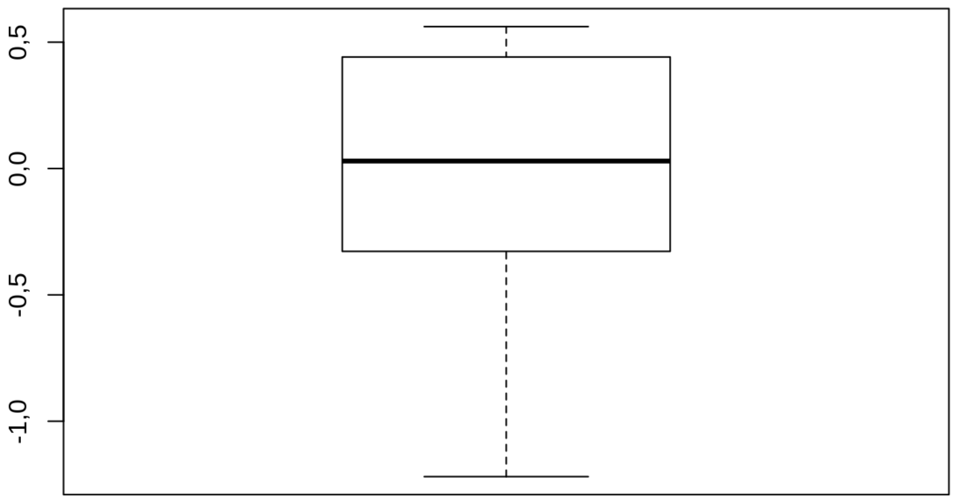
```
## `geom_smooth()` using formula 'y ~ x'
```



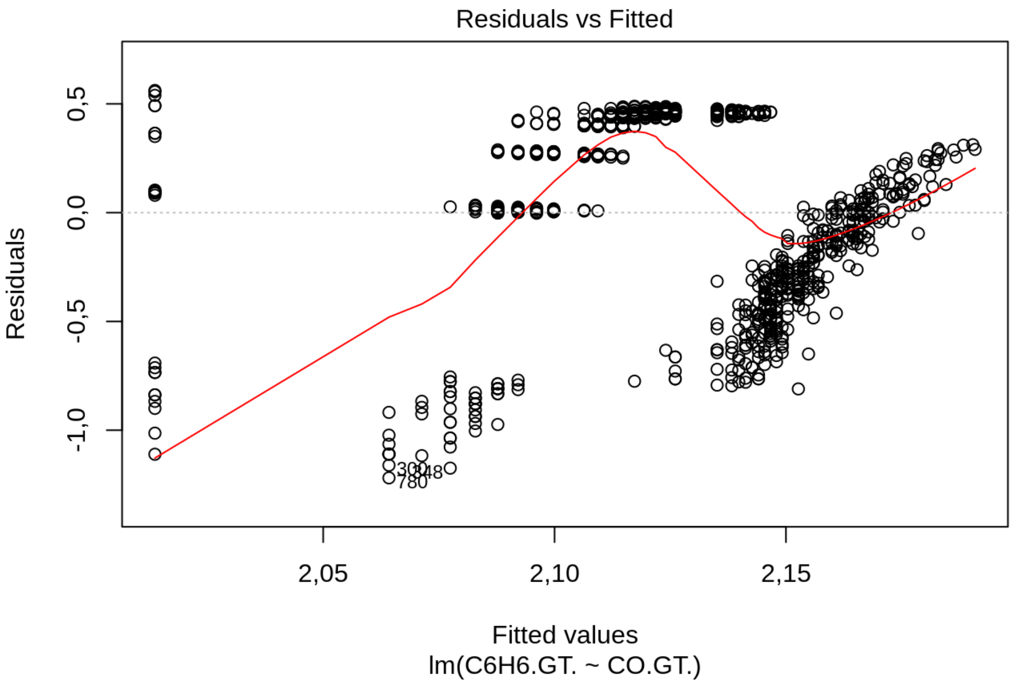

```
model_CO.GT. <- airq_data  %>%
  lm(data = .,  C6H6.GT. ~ CO.GT.)
residuals(model_CO.GT.) %>% hist()
```

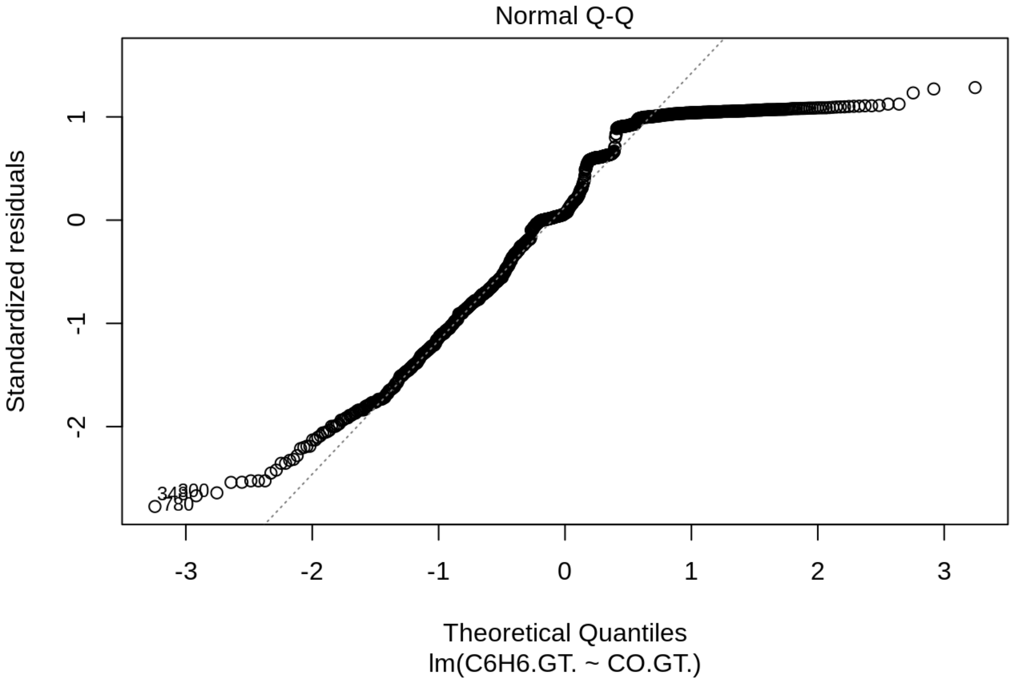



```
residuals(model_CO.GT.) %>% boxplot()
```




```
plot(model_CO.GT., which = c(1,2))
```






```
summary(model_CO.GT.)
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ CO.GT., data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1,21911 -0,32771  0,02954  0,44132  0,56155
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

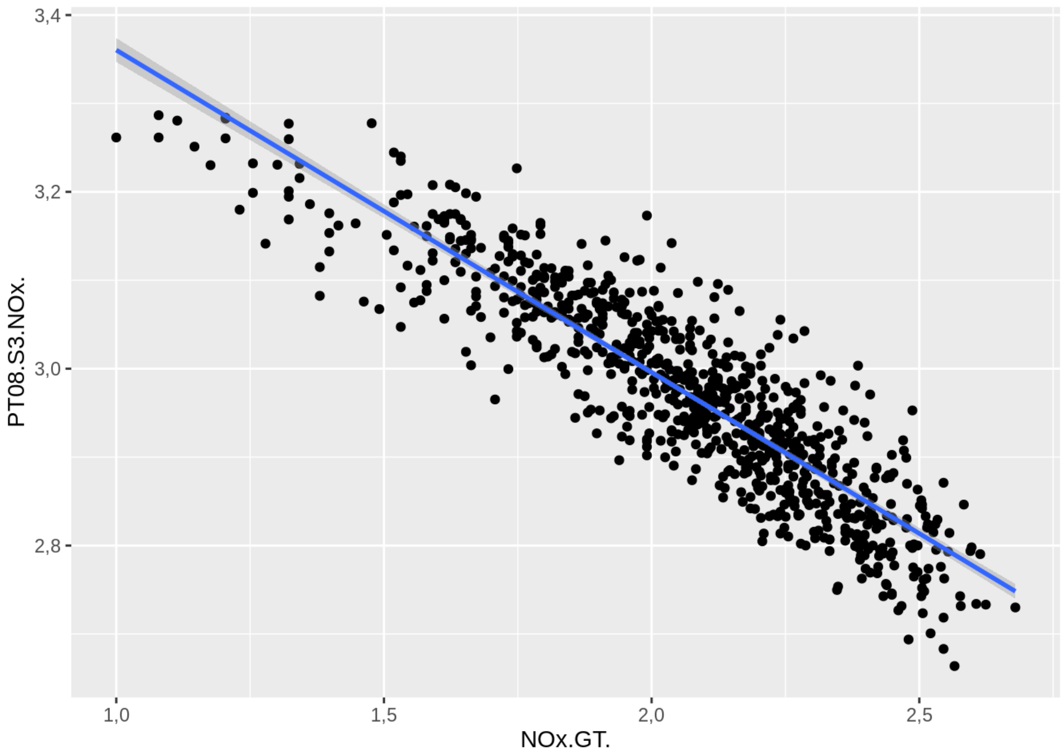
```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1,98172    0,06817  29,069   <2e-16 ***
## CO.GT.       0,10600    0,04815   2,202    0,028 *
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## Residual standard error: 0,4408 on 849 degrees of freedom
## Multiple R-squared:  0,005677,   Adjusted R-squared:  0,004506
## F-statistic: 4,847 on 1 and 849 DF,  p-value: 0,02796
```

So I decided to take a PT08.S3.NOx. as a predictor, because for me it seems a little bit better. I checked the assumptions for each variable and took 3 best:

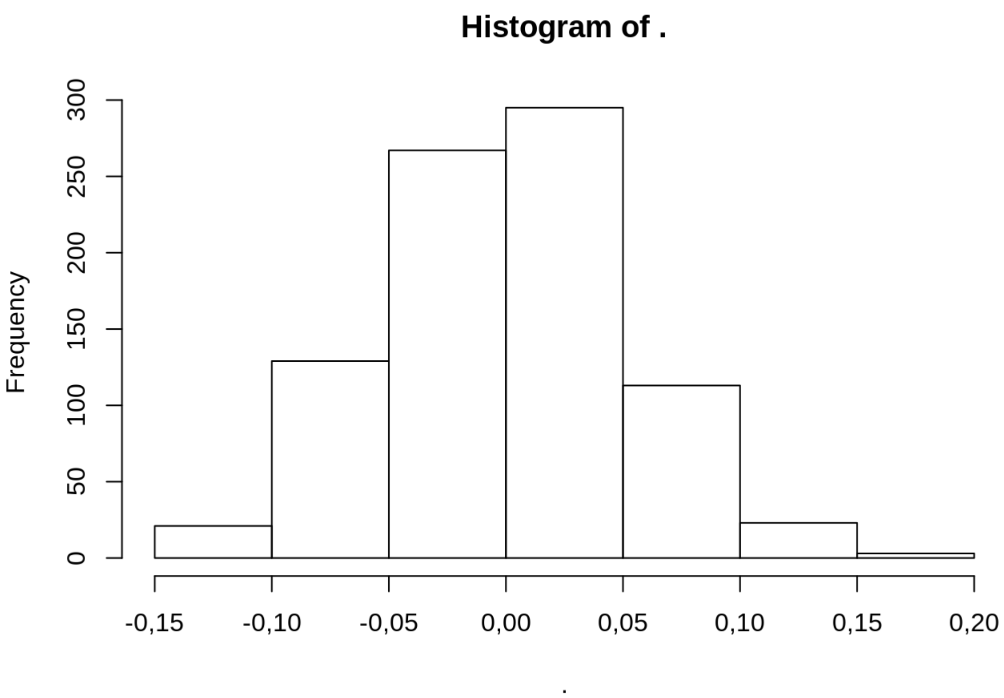
1. NOx.GT.

```
ggplot(airq_data, aes(x = NOx.GT., y = PT08.S3.NOx. )) +
  geom_point() +
  geom_smooth(method = 'lm')
```

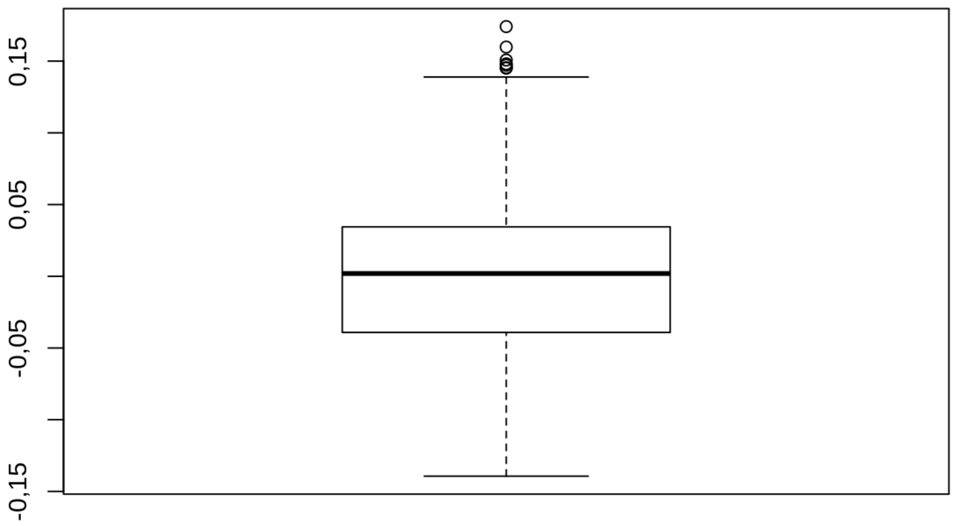
```
## `geom_smooth()` using formula 'y ~ x'
```



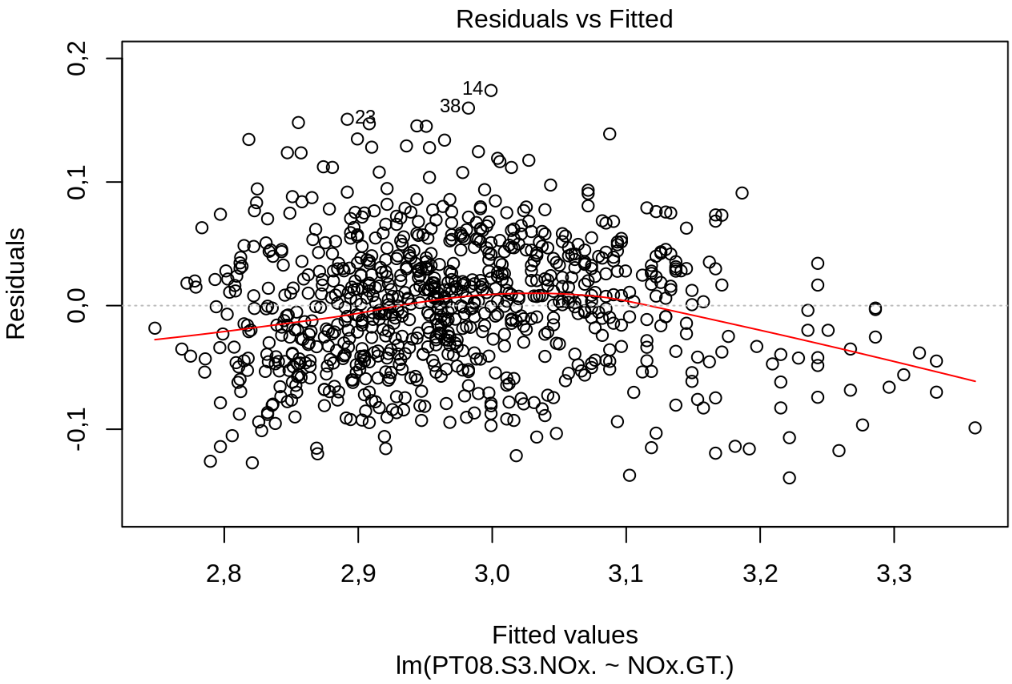

```
model_NOx.GT. <- airq_data %>% lm(data = ., PT08.S3.NOx. ~ NOx.GT.)
residuals(model_NOx.GT.) %>% hist()
```

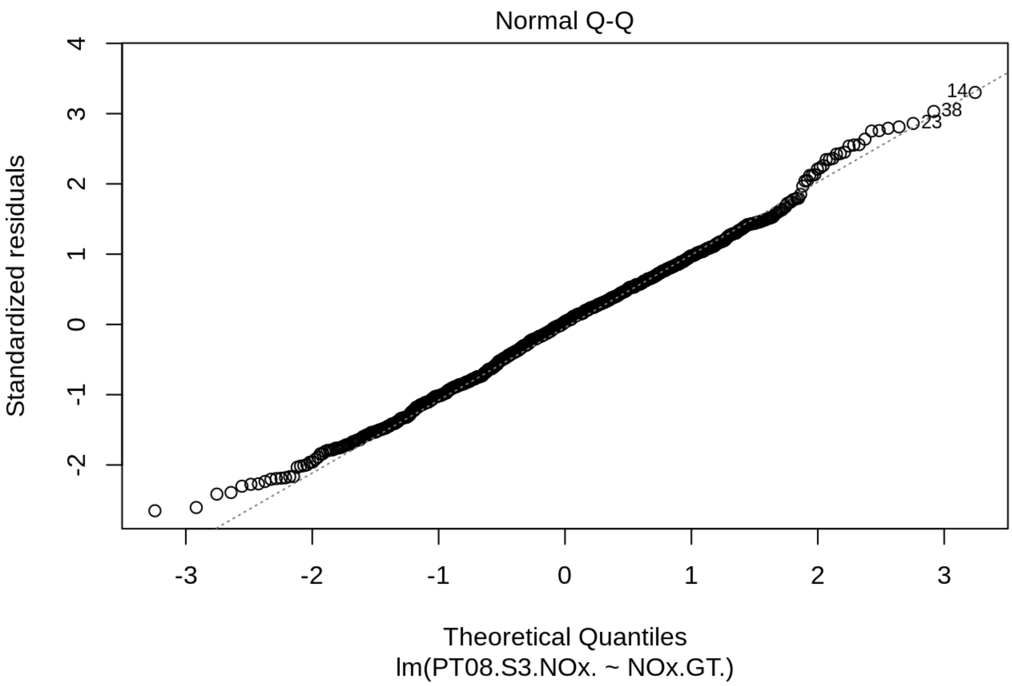



```
residuals(model_NOx.GT.) %>% boxplot()
```




```
plot(model_NOx.GT., which = c(1,2))
```






```
summary(model_NOx.GT.)
```

```
##
## Call:
## lm(formula = PT08.S3.NOx. ~ NOx.GT., data = .)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0,139384 -0,039168  0,001901  0,034434  0,174095
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3,724907   0,013022  286,04   <2e-16 ***
## NOx.GT.     -0,364507   0,006246  -58,36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## Residual standard error: 0,05276 on 849 degrees of freedom
## Multiple R-squared:  0,8005, Adjusted R-squared:  0,8002
## F-statistic:  3406 on 1 and 849 DF,  p-value: < 2,2e-16
```

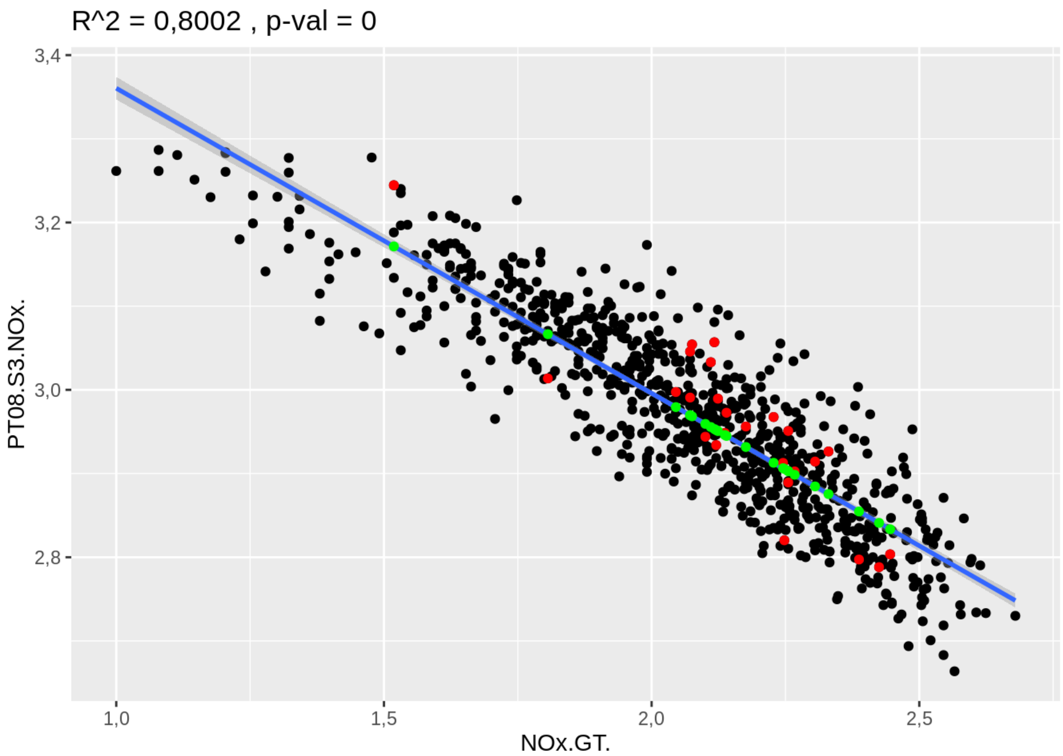
Prediction:

```
test_subset_NOx.GT. <-  airq_data[which(row.names(airq_data) %in% sample(row.names(airq_data), 25, replace = FALSE)), c(6,7)]
test_NOx.GT. <- data.frame(NOx.GT. = test_subset_NOx.GT.$NOx.GT.)
test_subset_NOx.GT.$pred_PT08.S3.NOx. <- predict(model_NOx.GT., newdata = test_NOx.GT.)
colnames(test_subset_NOx.GT.) <- c('real_NOx.GT.', 'real_PT08.S3.NOx.', 'pred_PT08.S3.NOx.')
head(test_subset_NOx.GT.)
```

```
##     real_NOx.GT. real_PT08.S3.NOx. pred_PT08.S3.NOx.
## 3       2,117271          3,056905          2,953147
## 16      2,110590          3,033021          2,955582
## 50      2,305351          2,914343          2,884590
## 77      2,330414          2,926342          2,875455
## 142     2,255273          2,950851          2,902844
## 292     2,255273          2,889302          2,902844
```

```
R <- round(summary(model_NOx.GT.)$adj.r.squared, digits = 4)
p <- round(summary(model_NOx.GT.)$coefficients[2,4], digits = 3)
titl <- paste('R^2 =', as.character(R),', p-val =', as.character(p))
ggplot() +
  geom_point(data = airq_data, aes(NOx.GT., PT08.S3.NOx.)) +
  geom_smooth(data = airq_data, aes(NOx.GT., PT08.S3.NOx.), method = 'lm') +
  geom_point(data = test_subset_NOx.GT., aes(real_NOx.GT., real_PT08.S3.NOx.), color = 'red') +
  geom_point(data = test_subset_NOx.GT., aes(real_NOx.GT., pred_PT08.S3.NOx.), color = 'green') +
  labs(title = titl)
```

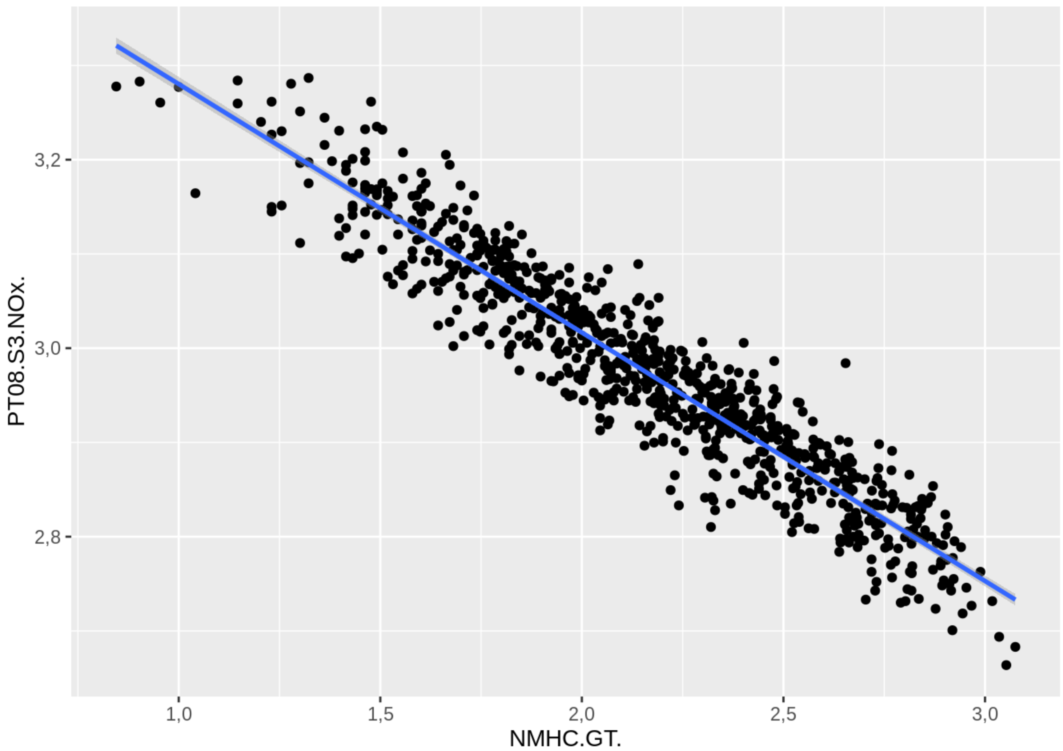
```
## `geom_smooth()` using formula 'y ~ x'
```



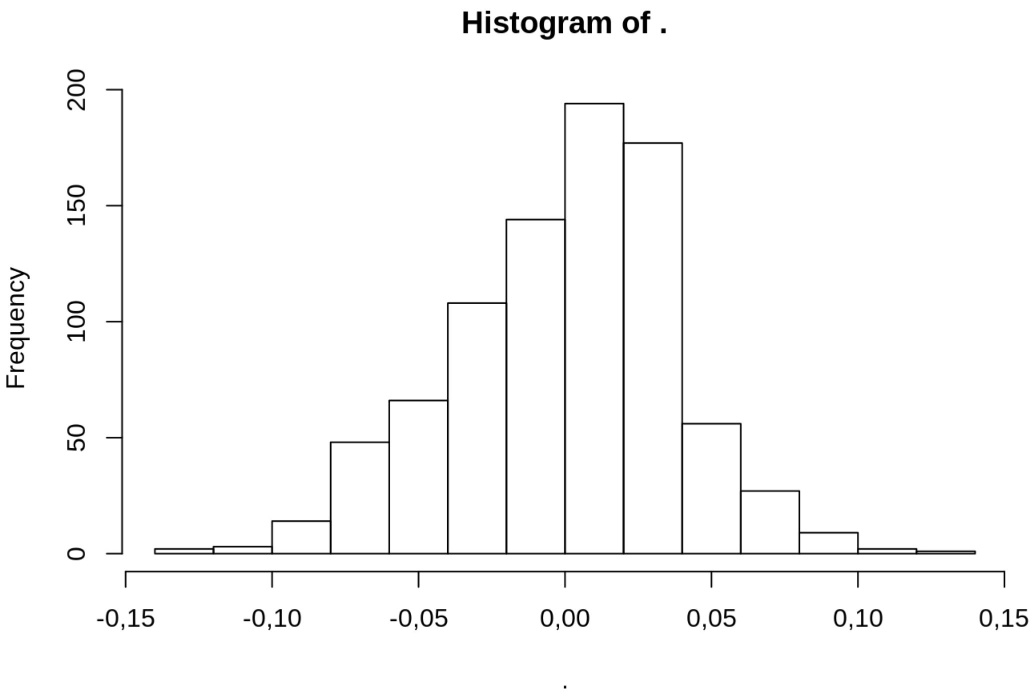
2. NMHC.GT.

```
ggplot(airq_data, aes(x = NMHC.GT., y = PT08.S3.NOx.)) +
  geom_point() +
  geom_smooth(method = 'lm')
```

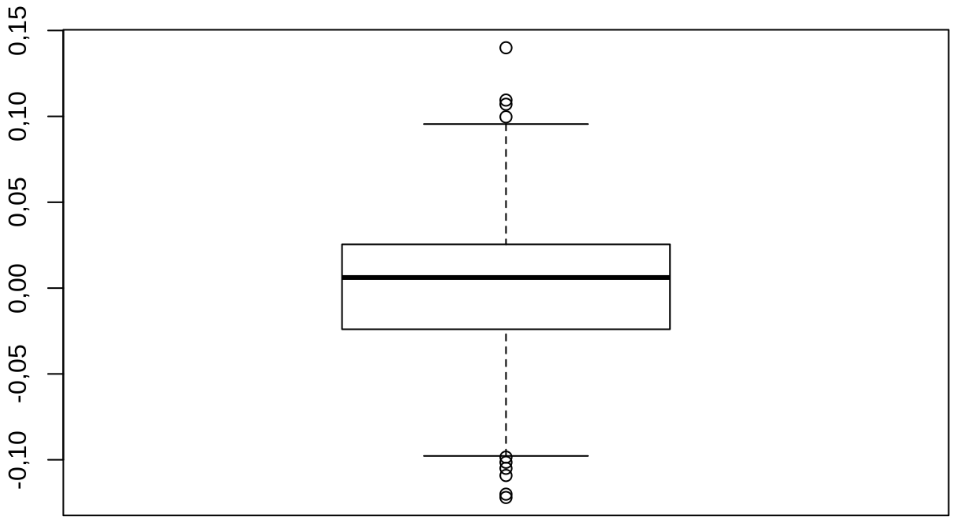
```
## `geom_smooth()` using formula 'y ~ x'
```



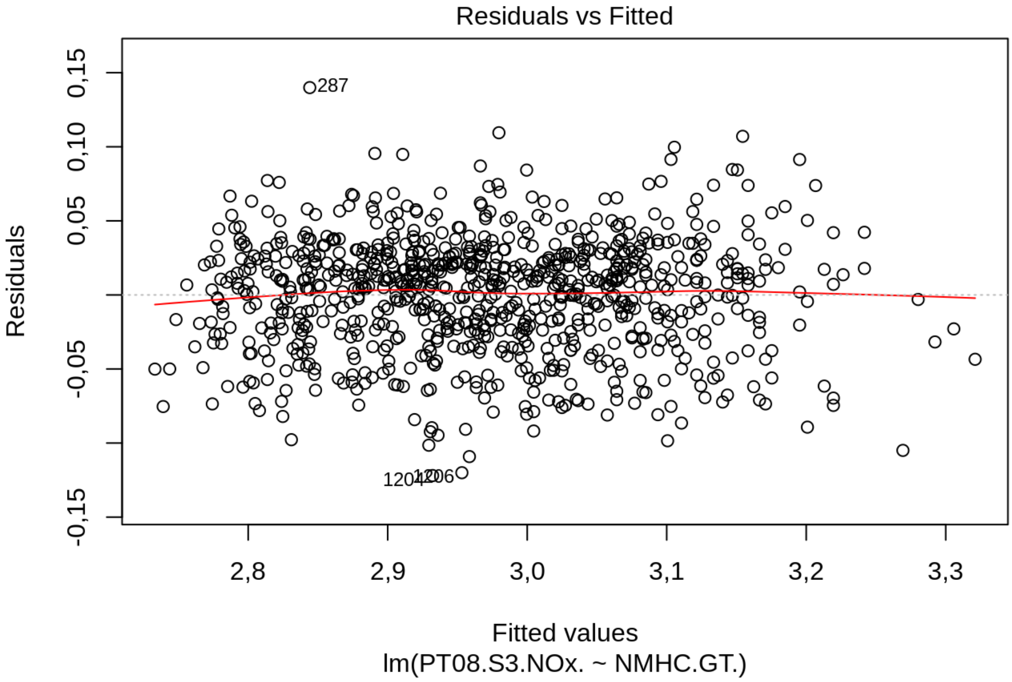
```
model_NMHC.GT. <- airq_data %>%  lm(data = ., PT08.S3.NOx.~ NMHC.GT.)
residuals(model_NMHC.GT.) %>% hist()
```

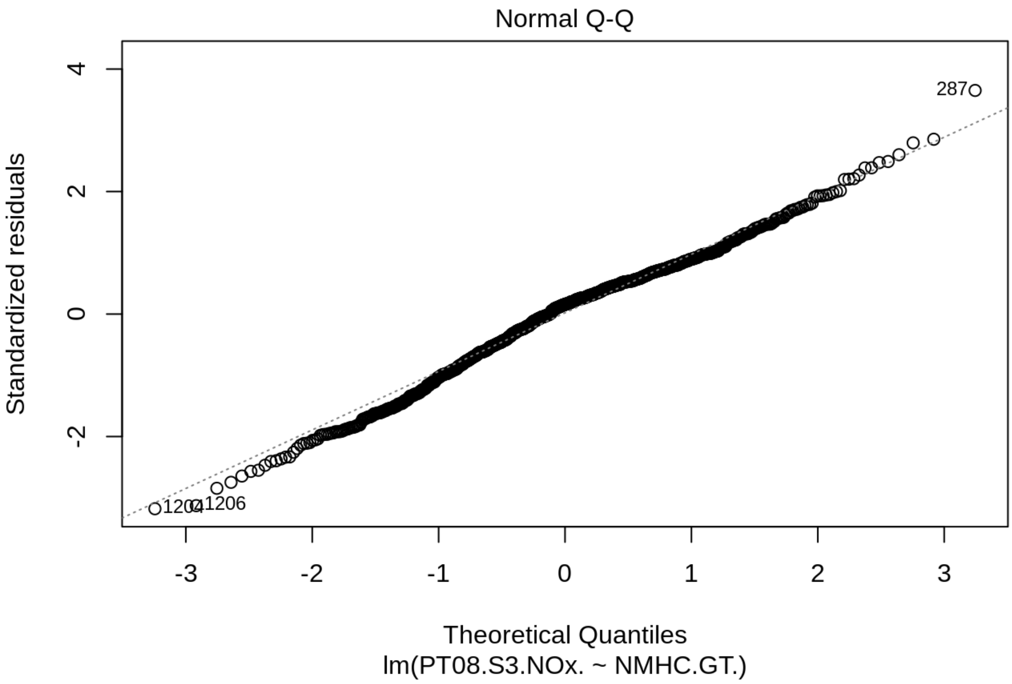



```
residuals(model_NMHC.GT.) %>% boxplot()
```




```
plot(model_NMHC.GT., which = c(1,2))
```






```
summary(model_NMHC.GT.)
```

```
##
## Call:
## lm(formula = PT08.S3.NOx. ~ NMHC.GT., data = .)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0,121954 -0,023971  0,006133  0,025483  0,139954
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3,543874   0,006867   516,0   <2e-16 ***
## NMHC.GT.    -0,263641   0,003109   -84,8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## Residual standard error: 0,03838 on 849 degrees of freedom
## Multiple R-squared:  0,8944, Adjusted R-squared:  0,8943
## F-statistic:  7191 on 1 and 849 DF,  p-value: < 2,2e-16
```

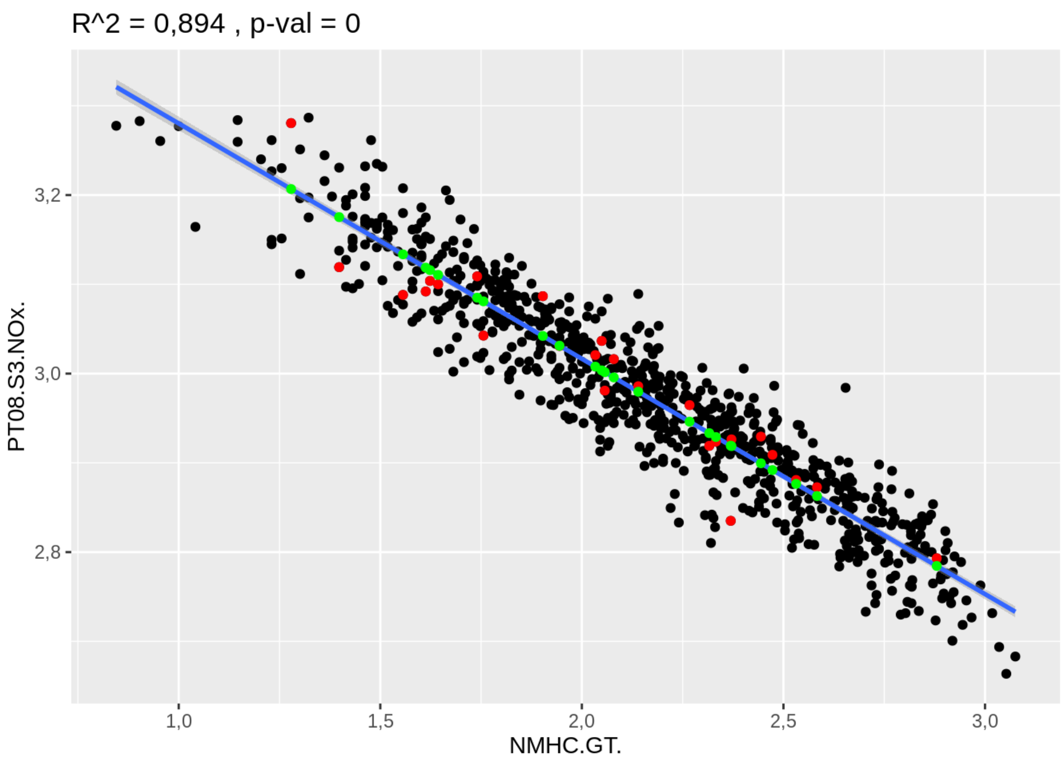
Prediction:

```
test_subset_NMHC.GT. <-  airq_data[which(row.names(airq_data) %in% sample(row.names(airq_data), 25, replace = FALSE)), c(3,7)]
test_NMHC.GT. <- data.frame(NMHC.GT. = test_subset_NMHC.GT.$NMHC.GT.)
test_subset_NMHC.GT.$pred_PT08.S3.NOx. <- predict(model_NMHC.GT., newdata = test_NMHC.GT.)
colnames(test_subset_NMHC.GT.) <- c('real_NMHC.GT.', 'real_PT08.S3.NOx.', 'pred_PT08.S3.NOx.')
head(test_subset_NMHC.GT.)
```

```
##     real_NMHC.GT. real_PT08.S3.NOx. pred_PT08.S3.NOx.
## 46       2,267172          2,964731          2,946153
## 53       2,332438          2,923762          2,928946
## 69       2,049218          3,036629          3,003615
## 70       2,033424          3,020775          3,007779
## 87       1,556303          3,088136          3,133568
## 116      2,315970          2,919078          2,933288
```

```
R <- round(summary(model_NMHC.GT.)$adj.r.squared, digits = 3)
p <- round(summary(model_NMHC.GT.)$coefficients[2,4], digits = 3)
titl <- paste('R^2 =', as.character(R),', p-val =', as.character(p))
ggplot() +
  geom_point(data = airq_data, aes(NMHC.GT., PT08.S3.NOx.)) +
  geom_smooth(data = airq_data, aes(NMHC.GT., PT08.S3.NOx.), method = 'lm') +
  geom_point(data = test_subset_NMHC.GT., aes(real_NMHC.GT., real_PT08.S3.NOx.), color = 'red') +
  geom_point(data = test_subset_NMHC.GT., aes(real_NMHC.GT., pred_PT08.S3.NOx.), color = 'green') +
  labs(title = titl)
```

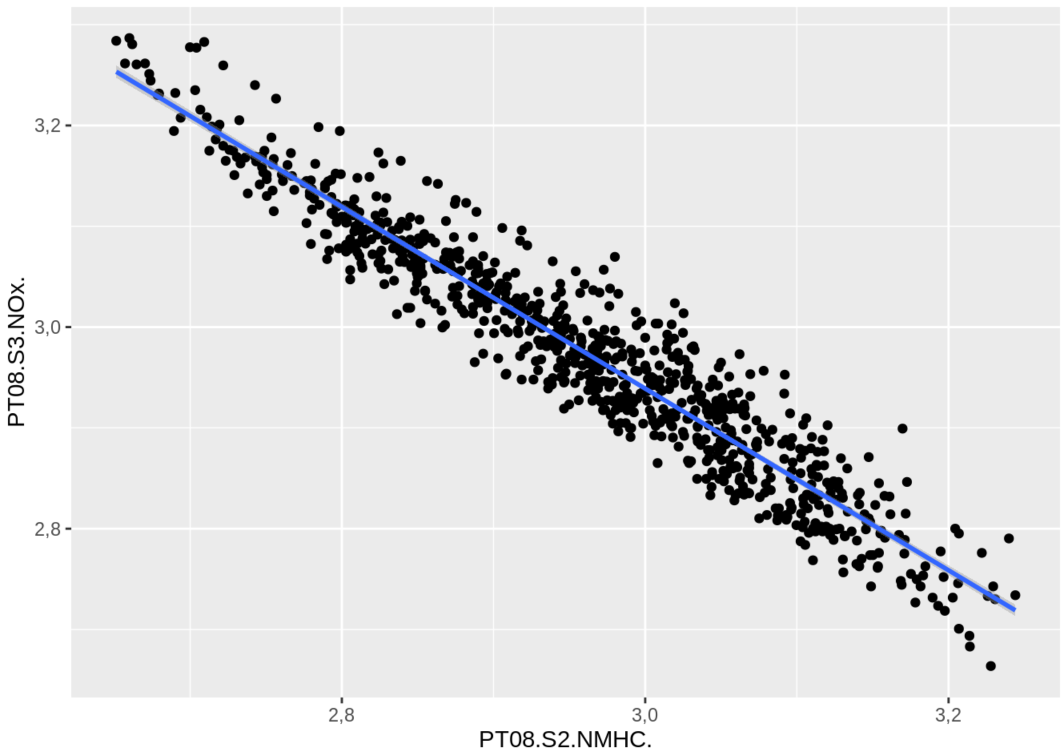
```
## `geom_smooth()` using formula 'y ~ x'
```



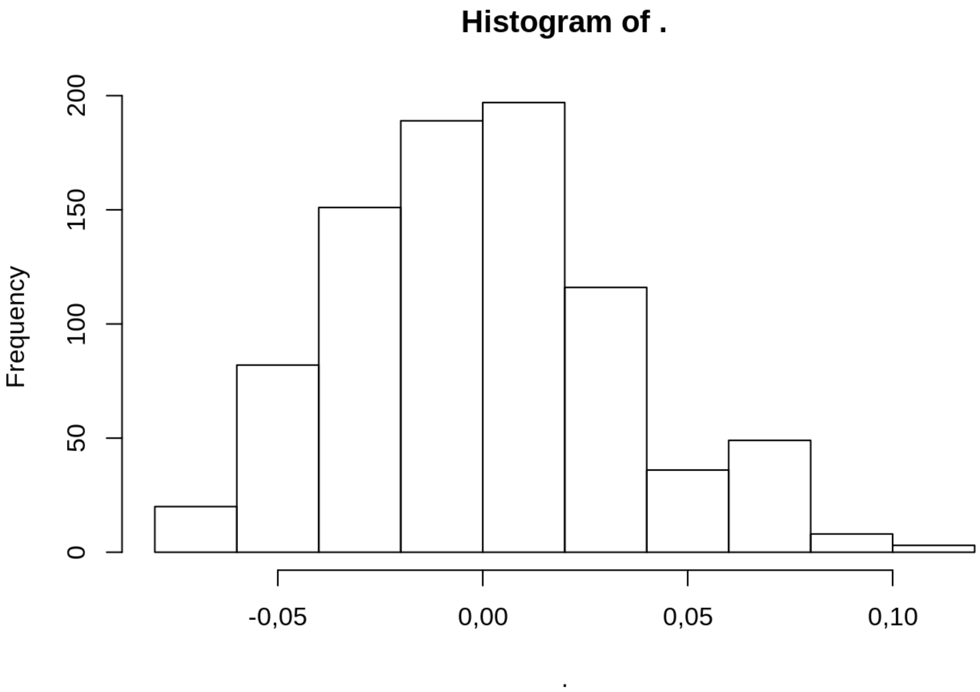
3. PT08.S2.NMHC.

```
ggplot(airq_data, aes(x = PT08.S2.NMHC., y = PT08.S3.NOx. )) +
  geom_point() +
  geom_smooth(method = 'lm')
```

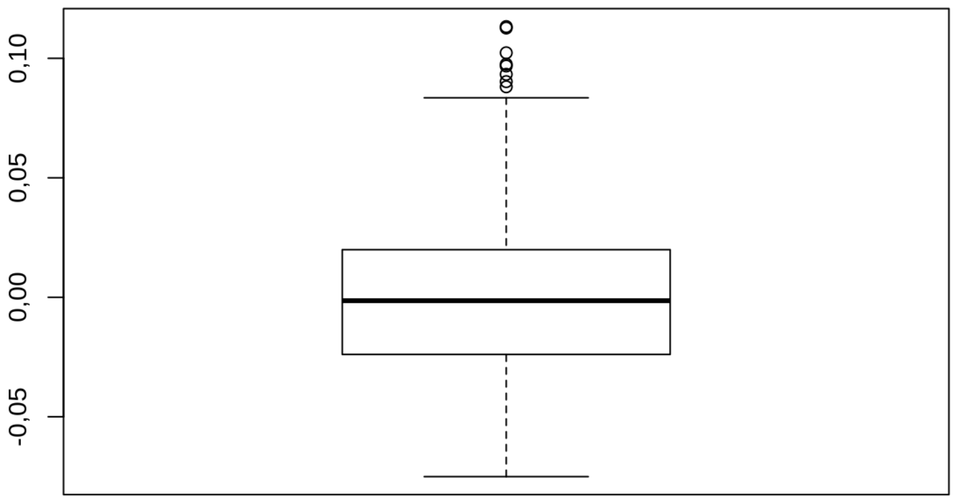
```
## `geom_smooth()` using formula 'y ~ x'
```



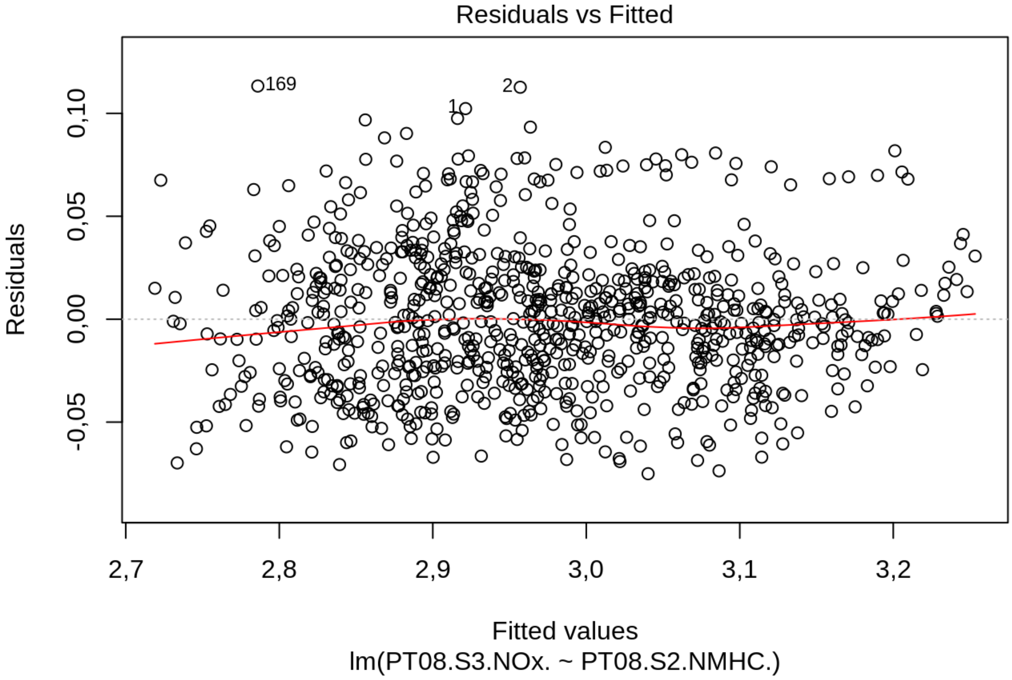

```
model_PT08.S2.NMHC. <- airq_data  %>% lm(data = .,  PT08.S3.NOx. ~ PT08.S2.NMHC.)
residuals(model_PT08.S2.NMHC.) %>% hist()
```

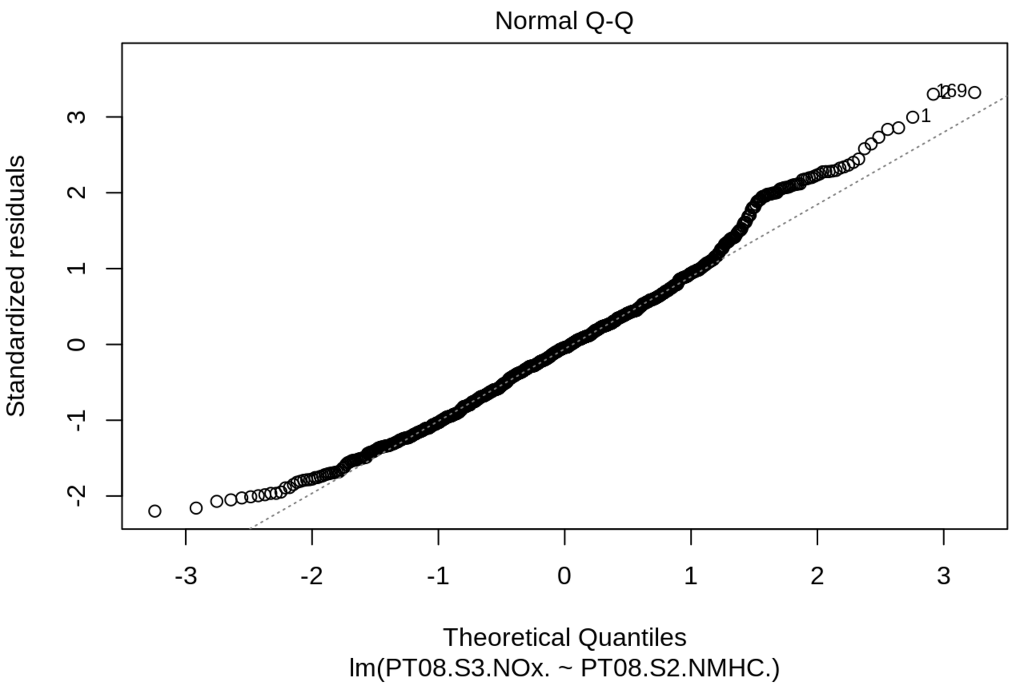



```
residuals(model_PT08.S2.NMHC.) %>% boxplot()
```




```r
plot(model_PT08.S2.NMHC., which = c(1,2))
```






```
summary(model_PT08.S2.NMHC.)
```

```
##
## Call:
## lm(formula = PT08.S3.NOx. ~ PT08.S2.NMHC., data = .)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0,075066 -0,023927 -0,001433  0,019915  0,113269
##
## Coefficients:
```

```
## Warning in printCoefmat(coefs, digits = digits, signif.stars = signif.stars, : в
## результате преобразования созданы NA
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5,64335    0,02773  203,53   <2e-16 ***
## PT08.S2.NMHC. -0,90146    0,00935  -96,42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## Residual standard error: 0,03417 on 849 degrees of freedom
## Multiple R-squared:  0,9163, Adjusted R-squared:  0,9162
## F-statistic:  9296 on 1 and 849 DF,  p-value: < 2,2e-16
```

Prediction:

```
test_subset_PT08.S2.NMHC. <-  airq_data[which(row.names(airq_data) %in% sample(row.names(airq_data), 25, replace = FALSE)), c(5,7)]
test_PT08.S2.NMHC. <- data.frame(PT08.S2.NMHC. = test_subset_PT08.S2.NMHC.$PT08.S2.NMHC.)
test_subset_PT08.S2.NMHC.$pred_PT08.S3.NOx. <- predict(model_PT08.S2.NMHC., newdata = test_PT08.S2.NMHC.)
colnames(test_subset_PT08.S2.NMHC.) <- c('real_PT08.S2.NMHC.', 'real_PT08.S3.NOx.', 'pred_PT08.S3.NOx.')
head(test_subset_PT08.S2.NMHC.)
```

```
##     real_PT08.S2.NMHC. real_PT08.S3.NOx. pred_PT08.S3.NOx.
## 76            3,095866          2,881955          2,852539
## 86            2,788875          3,141136          3,129280
## 89            2,965202          2,978637          2,970328
## 171           3,221936          2,775974          2,738892
## 180           2,794488          3,111599          3,124220
## 436           2,825426          3,100715          3,096331
```

```
R <- round(summary(model_PT08.S2.NMHC.)$adj.r.squared, digits = 3)
p <- round(summary(model_PT08.S2.NMHC.)$coefficients[2,4], digits = 3)
titl <- paste('R^2 =', as.character(R),', p-val =', as.character(p))
ggplot() +
  geom_point(data = airq_data, aes(PT08.S2.NMHC., PT08.S3.NOx.)) +
  geom_smooth(data = airq_data, aes(PT08.S2.NMHC., PT08.S3.NOx.), method = 'lm') +
  geom_point(data = test_subset_PT08.S2.NMHC., aes(real_PT08.S2.NMHC., real_PT08.S3.NOx.), color = 'red') +
  geom_point(data = test_subset_PT08.S2.NMHC., aes(real_PT08.S2.NMHC., pred_PT08.S3.NOx.), color = 'green') +
  labs(title = titl)
```

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