Task_2.1

NataliaBaymacheva

5/26/2020

Loading libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(tidyr)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

Look at the data

```
df <- anscombe
head(df)
```

```
##   x1 x2 x3 x4   y1   y2    y3   y4
## 1 10 10 10  8 8.04 9.14  7.46 6.58
## 2  8  8  8  8 6.95 8.14  6.77 5.76
## 3 13 13 13  8 7.58 8.74 12.74 7.71
## 4  9  9  9  8 8.81 8.77  7.11 8.84
## 5 11 11 11  8 8.33 9.26  7.81 8.47
## 6 14 14 14  8 9.96 8.10  8.84 7.04
```

Converting the data to a better format

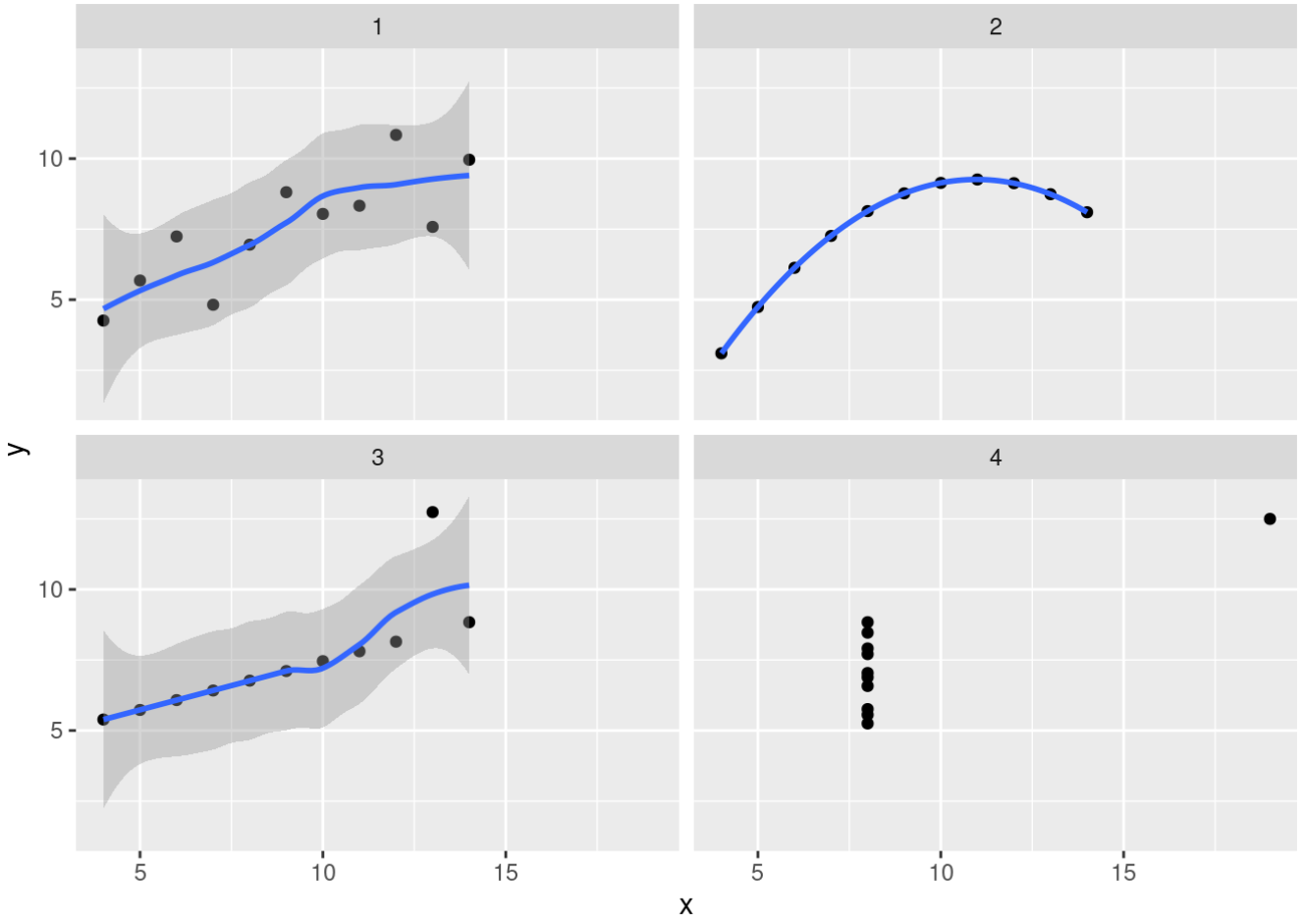
```
df1 <- data.frame(cbind(df$x1, df$y1, 'set' = 1))
df2 <- data.frame(cbind(df$x2, df$y2, 'set' = 2))
df3 <- data.frame(cbind(df$x3, df$y3, 'set' = 3))
df4 <- data.frame(cbind(df$x4, df$y4, 'set' = 4))
df <- rbind(df1, df2, df3, df4)
names(df)[names(df) == 'V1'] <- 'x'
names(df)[names(df) == 'V2'] <- 'y'

head(df)
```

```
##    x    y set
## 1 10 8.04   1
## 2  8 6.95   1
## 3 13 7.58   1
## 4  9 8.81   1
## 5 11 8.33   1
## 6 14 9.96   1
```

Plotting the data

```
df %>%
  ggplot(aes(x = x, y = y)) +
  geom_point() +
  facet_wrap(~set) +
  geom_smooth()
```



Mean/sd summary table

```
df %>%
  group_by(set) %>%
  summarise(mean_x=mean(x), sd_x = sd(x), mean_y=mean(y), sd_y = sd(y))
```

```
## # A tibble: 4 x 5
##     set mean_x  sd_x mean_y  sd_y
##   <dbl>  <dbl> <dbl>  <dbl> <dbl>
```

```
## 1     1       9  3.32   7.50  2.03
## 2     2       9  3.32   7.50  2.03
## 3     3       9  3.32   7.5   2.03
## 4     4       9  3.32   7.50  2.03
```

Pearson's/non-parametric summary table

```
df %>%
  group_by(set) %>%
  summarise(pearson_est=cor.test(x, y)$estimate, pearson_p=cor.test(x, y)$p.value,
            kendall_est=cor.test(x, y, method = 'kendall')$estimate, kendall_p=cor.test(x, y, method = 'kendall')
$p.value,
            spearman_est=cor.test(x, y, method = 'spearman')$estimate, spearman_p=cor.test(x, y, method = 'spearm
an')$p.value)
```

```
## # A tibble: 4 x 7
##     set pearson_est pearson_p kendall_est   kendall_p spearman_est spearman_p
##   <dbl>       <dbl>     <dbl>       <dbl>       <dbl>        <dbl>      <dbl>
## 1     1       0.816   0.00217       0.636 0.00571             0.818    0.00373
## 2     2       0.816   0.00218       0.564 0.0165              0.691    0.0231
## 3     3       0.816   0.00218       0.964 0.000000551         0.991    0
## 4     4       0.817   0.00216       0.426 0.114               0.5      0.117
```

```
rm(list = ls())
```

AIRQUALITY DATA

Loading libraries

```
library(dplyr)
library(ggplot2)
library(tidyr)
library(corrplot)
```

```
airquality_data <- read.csv2('~/R/AirQualityUCI.csv', sep = ';', blank.lines.skip = T)[,1:15]
head(airquality_data)
```

```
##         Date     Time CO.GT. PT08.S1.CO. NMHC.GT. C6H6.GT. PT08.S2.NMHC.
## 1 10/03/2004 18.00.00    2.6        1360      150     11.9           1046
## 2 10/03/2004 19.00.00    2.0        1292      112      9.4            955
## 3 10/03/2004 20.00.00    2.2        1402       88      9.0            939
## 4 10/03/2004 21.00.00    2.2        1376       80      9.2            948
## 5 10/03/2004 22.00.00    1.6        1272       51      6.5            836
## 6 10/03/2004 23.00.00    1.2        1197       38      4.7            750
##   NOx.GT. PT08.S3.NOx. NO2.GT. PT08.S4.NO2. PT08.S5.O3.    T   RH     AH
## 1     166         1056     113         1692        1268 13.6 48.9 0.7578
## 2     103         1174      92         1559         972 13.3 47.7 0.7255
## 3     131         1140     114         1555        1074 11.9 54.0 0.7502
## 4     172         1092     122         1584        1203 11.0 60.0 0.7867
## 5     131         1205     116         1490        1110 11.2 59.6 0.7888
## 6      89         1337      96         1393         949 11.2 59.2 0.7848
```

There are many NA lines, get rid of them

```
airquality_data <- drop_na(airquality_data)
```

Transforming dates to date types

```
airquality_data$Date <- as.Date(airquality_data$Date)
```

Checcking the structures of the cols

```
str(airquality_data)
```

```
## 'data.frame':    9357 obs. of  15 variables:
##  $ Date         : Date, format: "10-03-20" "10-03-20" ...
##  $ Time         : Factor w/ 25 levels "","00.00.00",..: 20 21 22 23 24 25 2 3 4 5 ...
##  $ CO.GT.       : num  2.6 2 2.2 2.2 1.6 1.2 1.2 1 0.9 0.6 ...
```

```
##  $ 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.     : num  11.9 9.4 9 9.2 6.5 4.7 3.6 3.3 2.3 1.7 ...
##  $ 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            : 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 ...
```

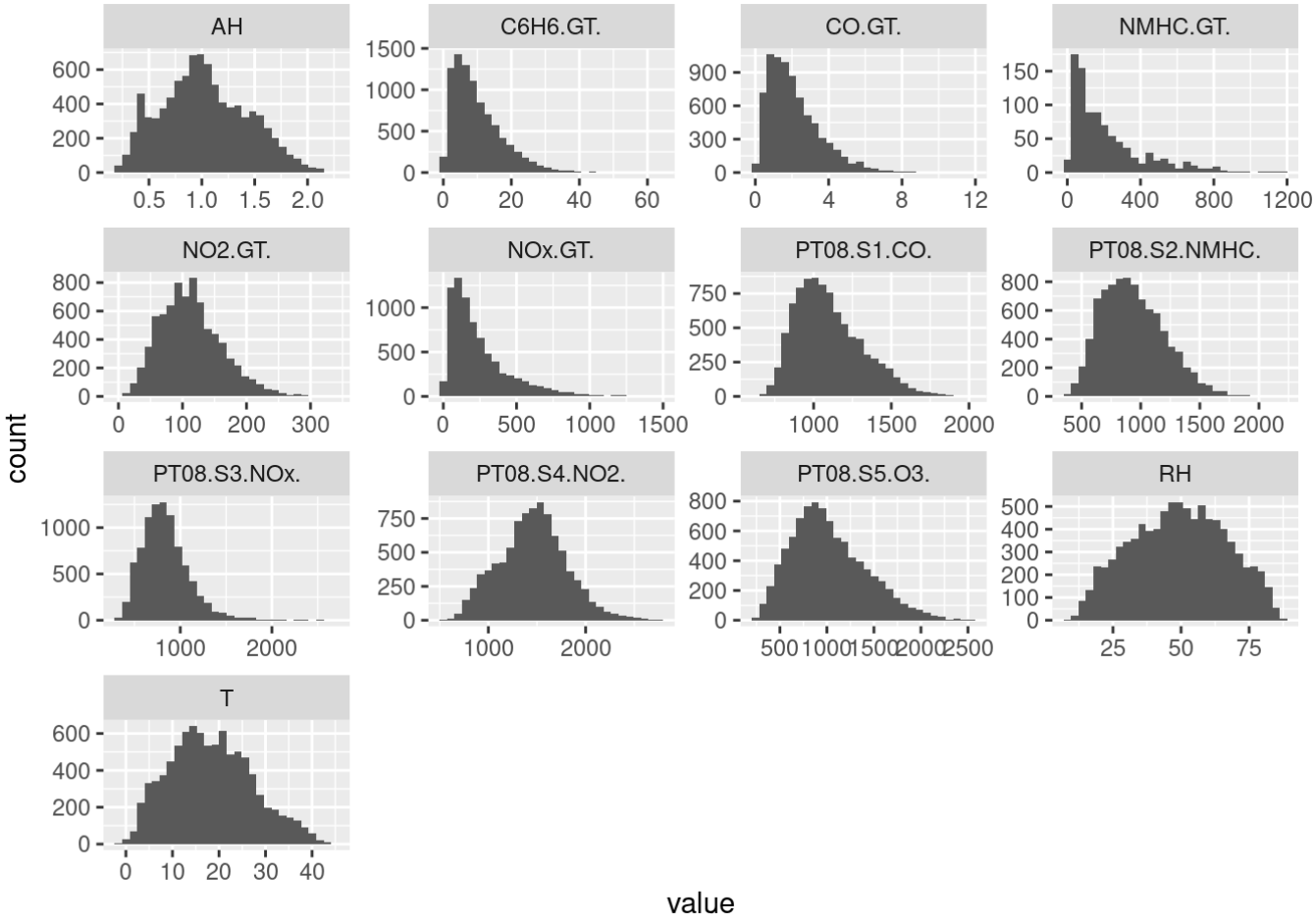
Change all -200 to na

```
for (col in 3:15) {
  airquality_data[which(airquality_data[col] == -200), col] <- NA
}
```

Check distributions

```
airquality_long <- pivot_longer(airquality_data, cols = 3:15, names_to = 'measure', values_to = 'value', values_d
rop_na = T)
airquality_long %>%
  group_by(measure) %>%
  ggplot(aes(x = value)) +
  geom_histogram() +
  facet_wrap(~measure, scales = 'free')
```

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



Not all data is normalli distributed and all variables have different scaling

Perform data normalization

```
airquality_norm_long <- airquality_long
airquality_norm_long$value <- log10(airquality_norm_long$value)
```

```
## Warning: NaNs produced
```

```
airqual_norm <- airquality_data[, 3:15]
airqual_norm <- log10(airqual_norm)
```

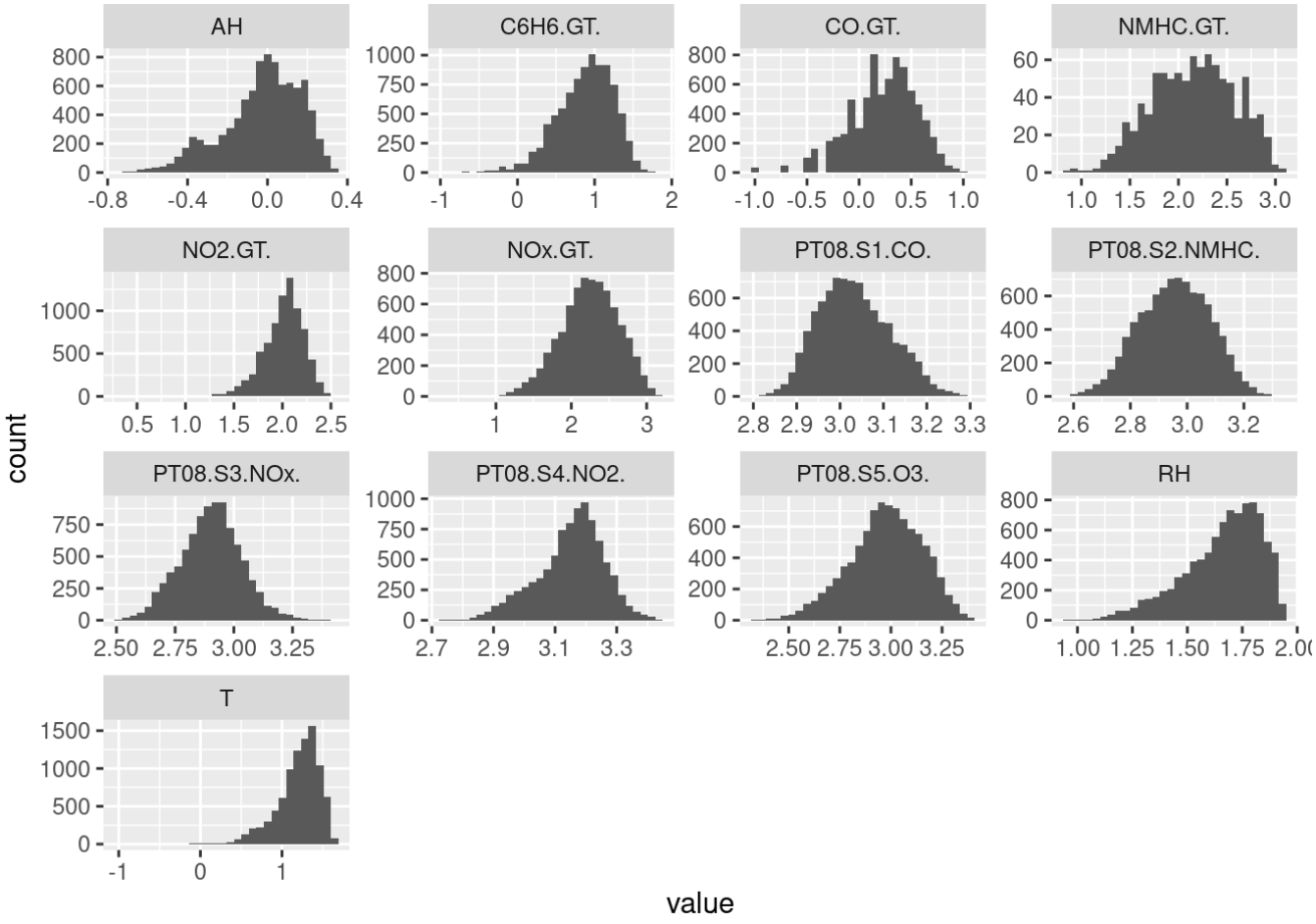
```
## Warning in lapply(X = x, FUN = .Generic, ...): NaNs produced
```

Plot normilized data

```
airquality_norm_long %>%
  group_by(measure) %>%
  ggplot(aes(x = value)) +
  geom_histogram() +
  facet_wrap(~measure, scales = 'free')
```

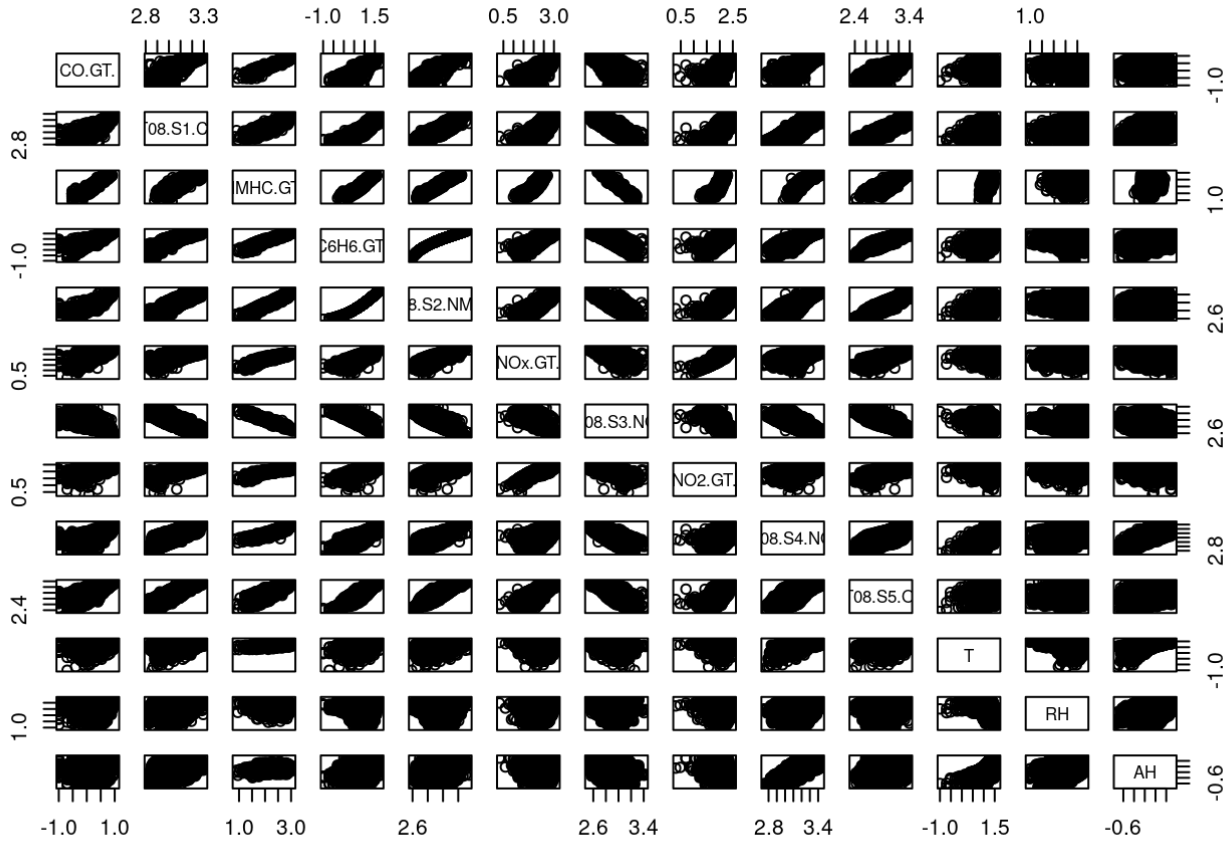
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 14 rows containing non-finite values (stat_bin).
```



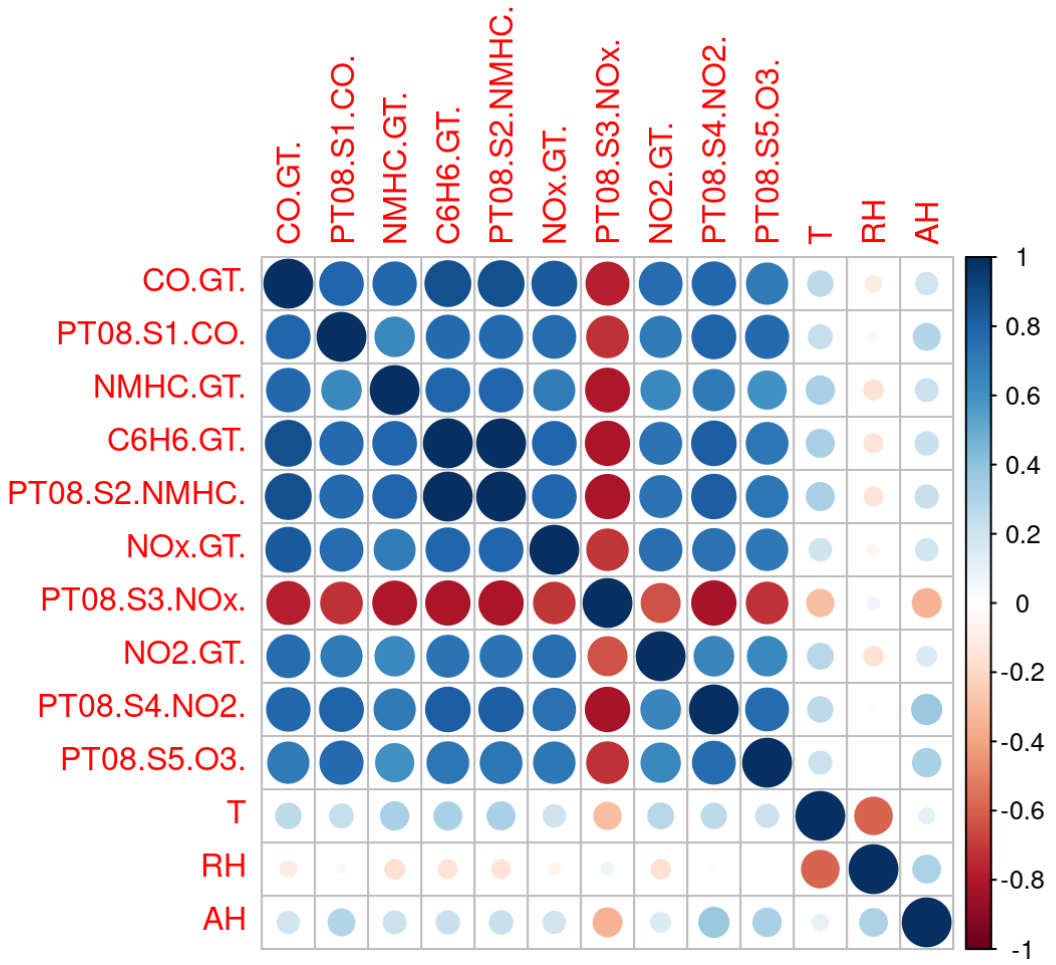
Plotting each pair of variables

```
pairs(airqual_norm)
```

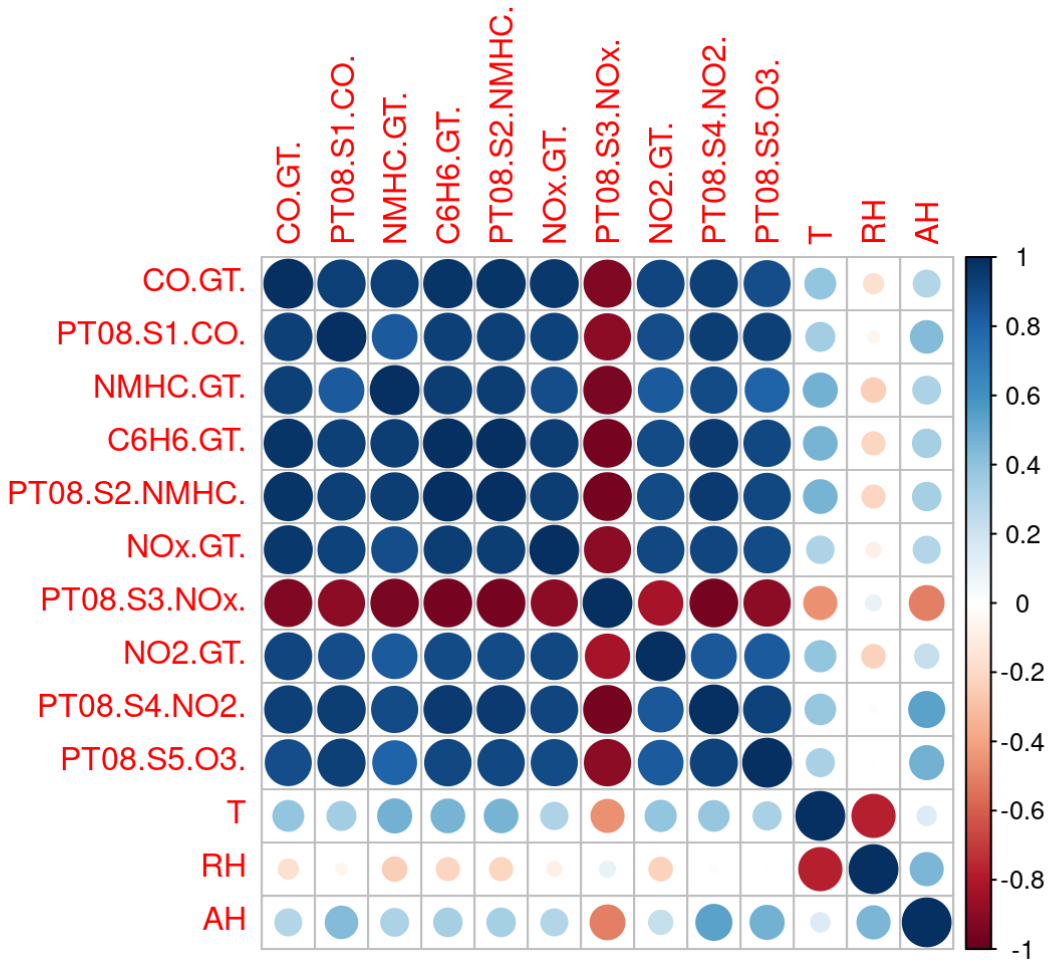


Cross-correlation

```
corr_kendall <- cor(airquality_data[, 3:15], use = 'complete.obs', method = 'kendall')
corrplot(corr_kendall)
```




```
corr_spearman <- cor(airquality_data[, 3:15], use = 'complete.obs', method = 'spearman')
corrplot(corr_spearman)
```



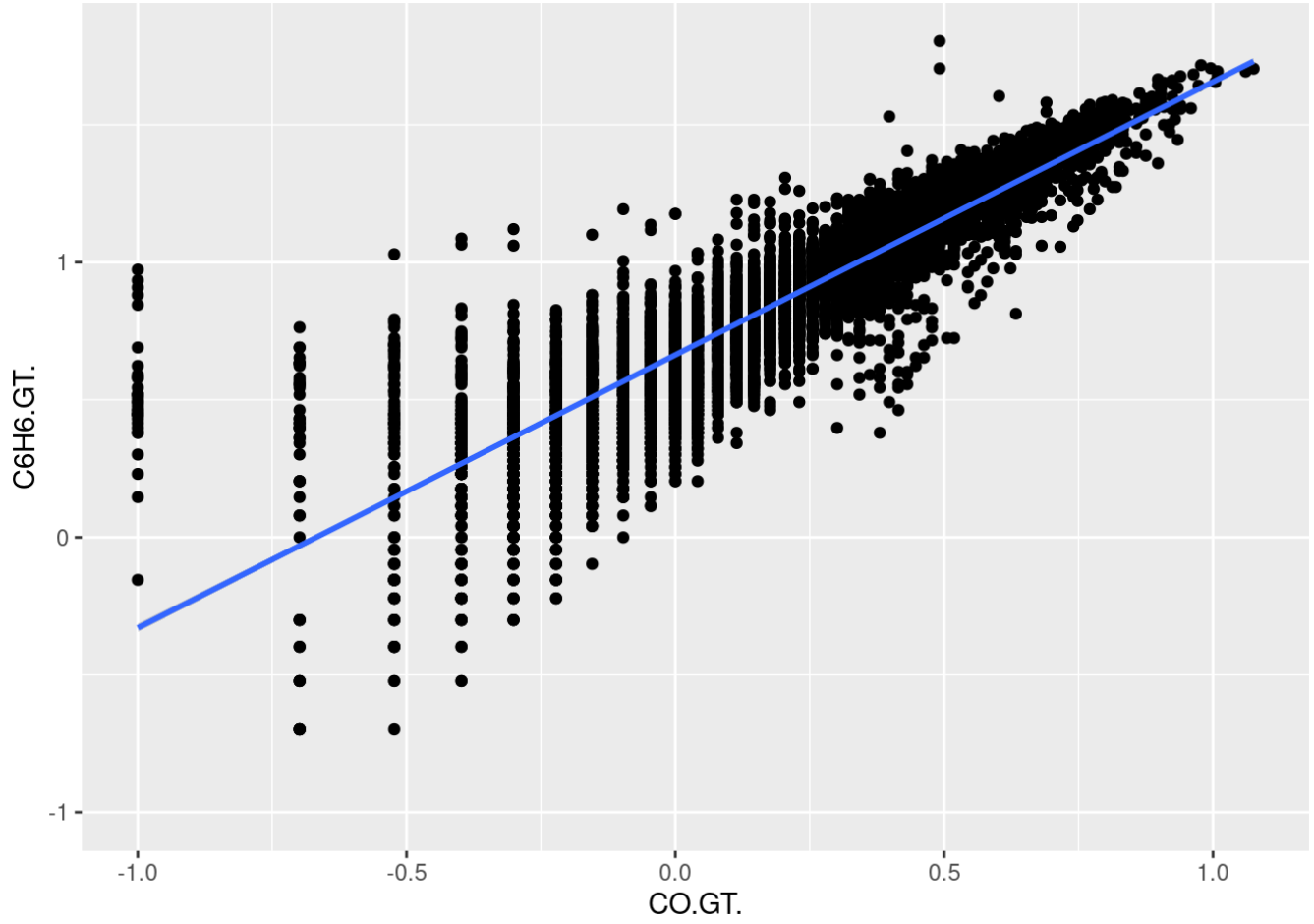
Looks like spearman method detects stronger correlation between benzene and other variables.

Plotting Response of benzene for each predictor

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

```
## Warning: Removed 2013 rows containing non-finite values (stat_smooth).
```

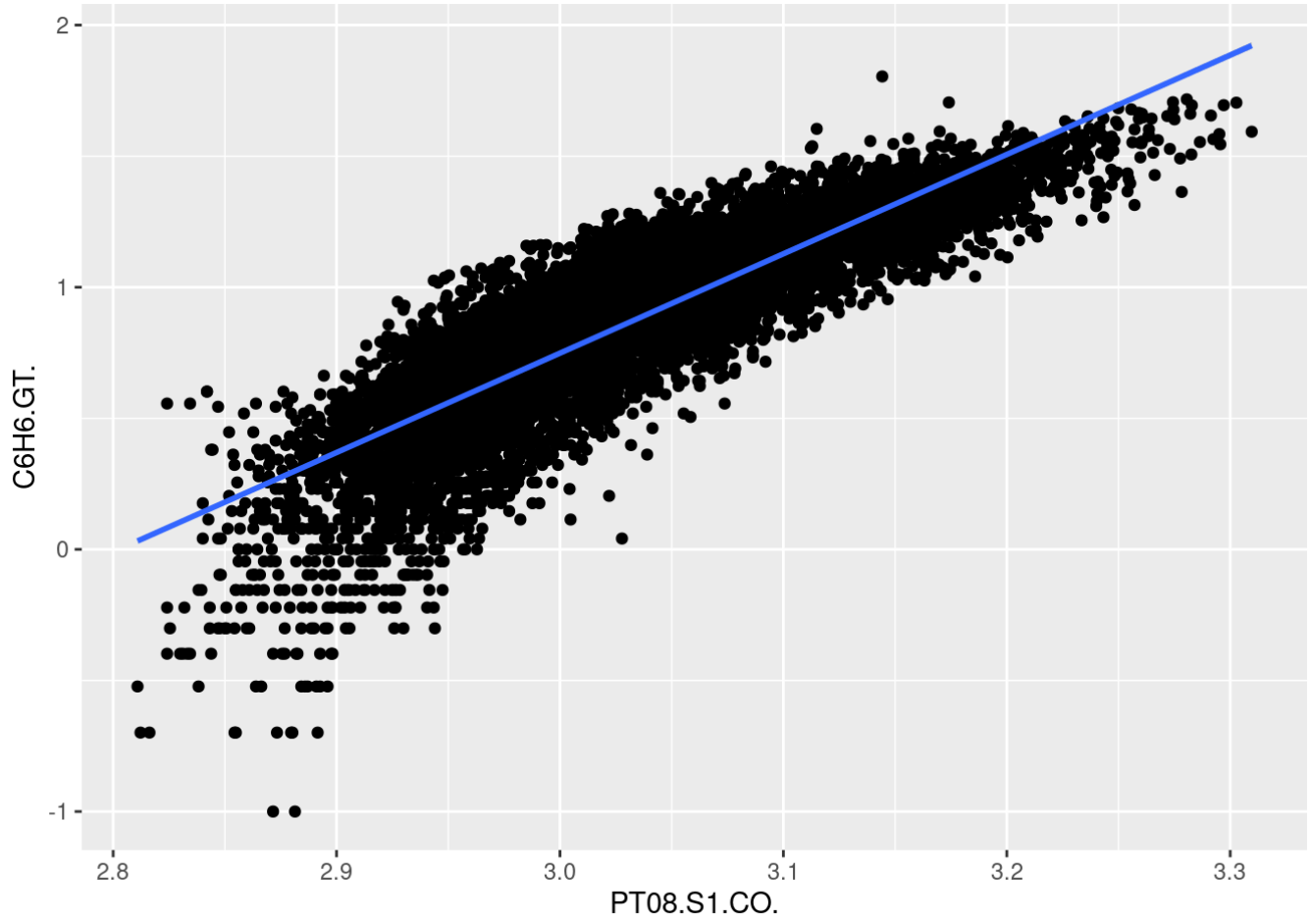
```
## Warning: Removed 2013 rows containing missing values (geom_point).
```




```
airqual_norm %>%
  ggplot(aes(x = PT08.S1.CO., y = C6H6.GT.)) +
  geom_point() +
  geom_smooth(method = 'lm')
```

```
## Warning: Removed 366 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 366 rows containing missing values (geom_point).
```

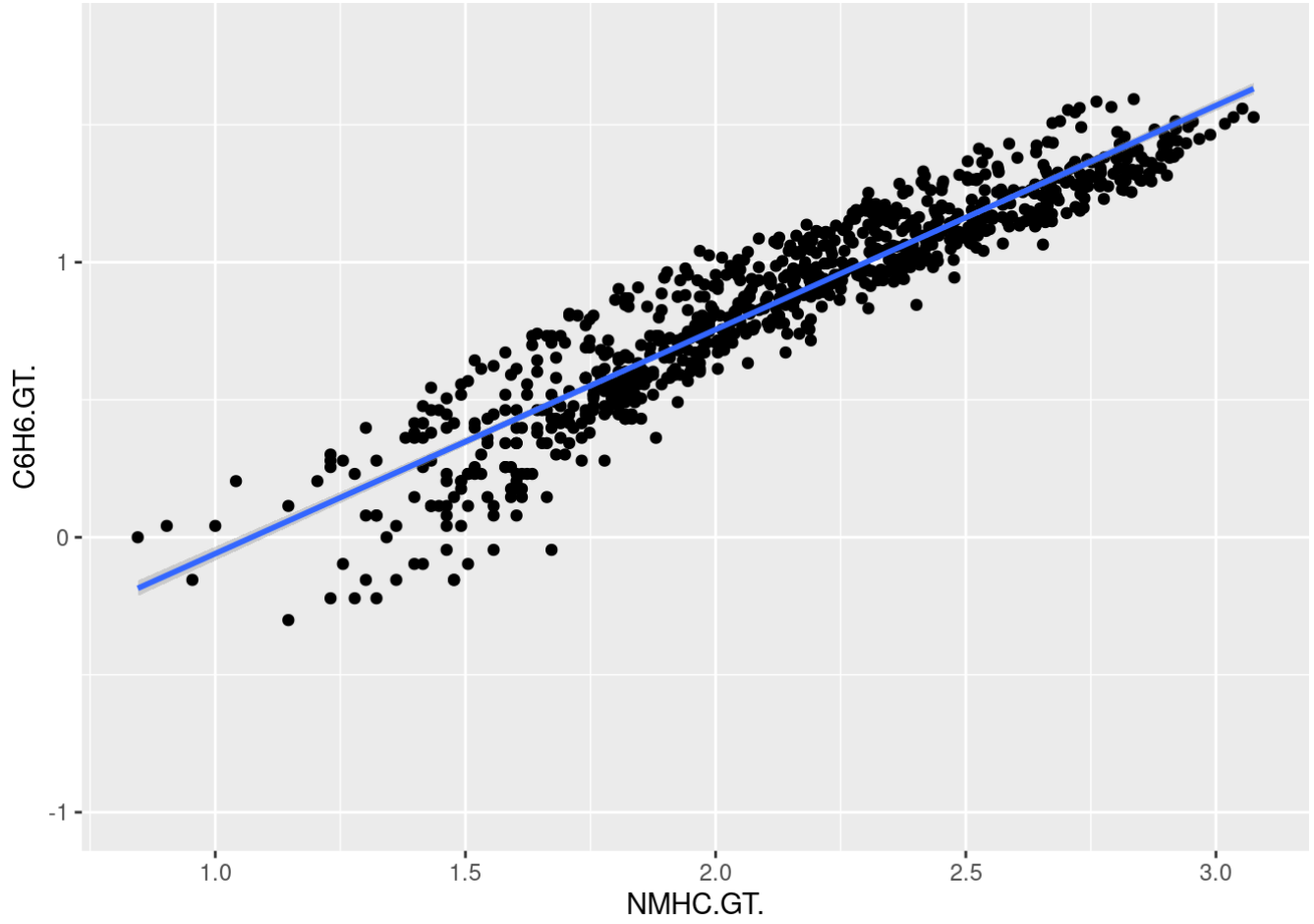



```
airqual_norm %>%
  ggplot(aes(x = NMHC.GT., y = C6H6.GT.)) +
```

```
  geom_point() +
  geom_smooth(method = 'lm')
```

```
## Warning: Removed 8470 rows containing non-finite values (stat_smooth).
```

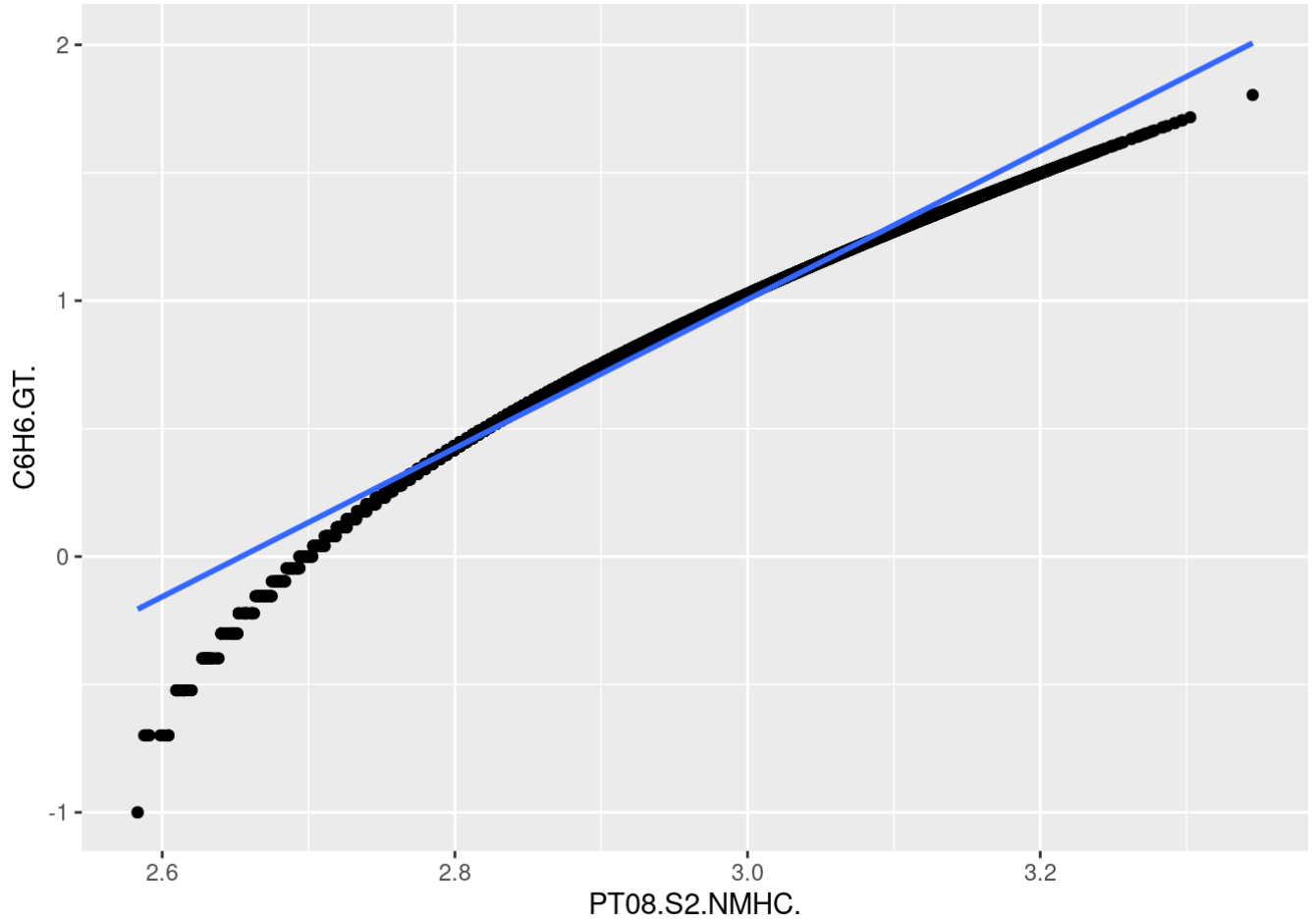
```
## Warning: Removed 8470 rows containing missing values (geom_point).
```



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

```
## Warning: Removed 366 rows containing non-finite values (stat_smooth).
```

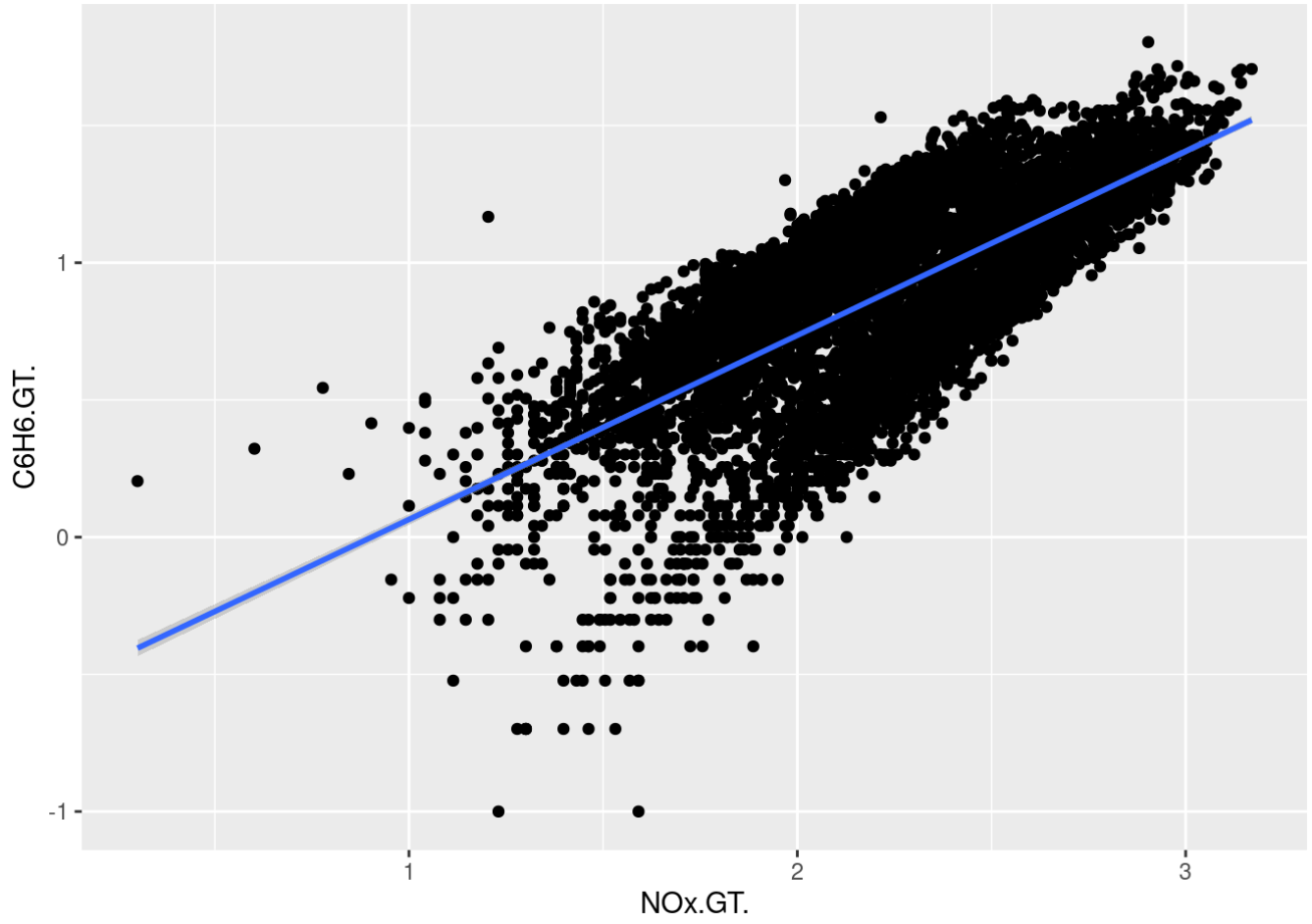
```
## Warning: Removed 366 rows containing missing values (geom_point).
```




```
airqual_norm %>%
  ggplot(aes(x = NOx.GT., y = C6H6.GT.)) +
  geom_point() +
  geom_smooth(method = 'lm')
```

```
## Warning: Removed 1961 rows containing non-finite values (stat_smooth).
```

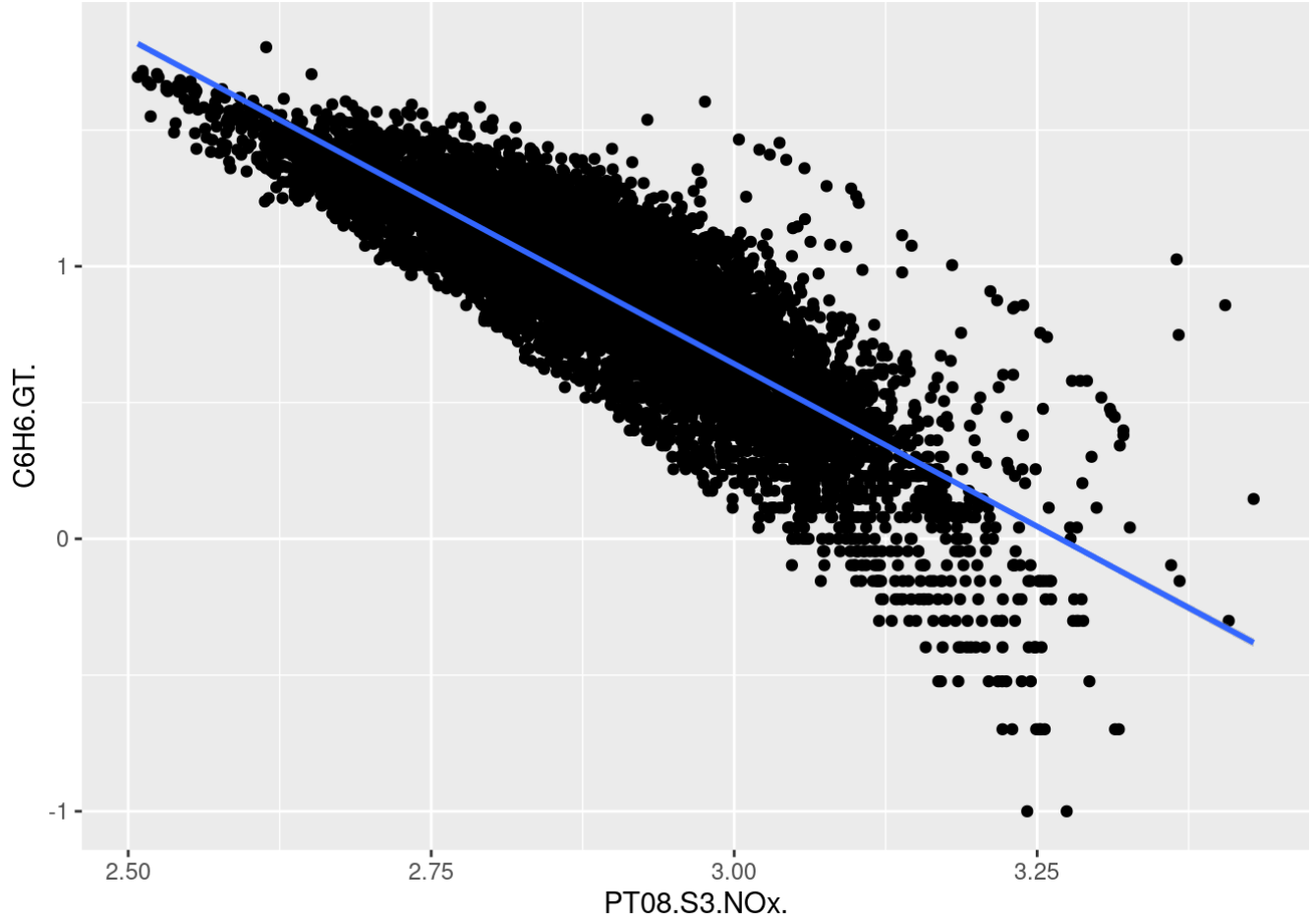
```
## Warning: Removed 1961 rows containing missing values (geom_point).
```




```
airqual_norm %>%
  ggplot(aes(x = PT08.S3.NOx., y = C6H6.GT.)) +
  geom_point() +
  geom_smooth(method = 'lm')
```

```
## Warning: Removed 366 rows containing non-finite values (stat_smooth).
```

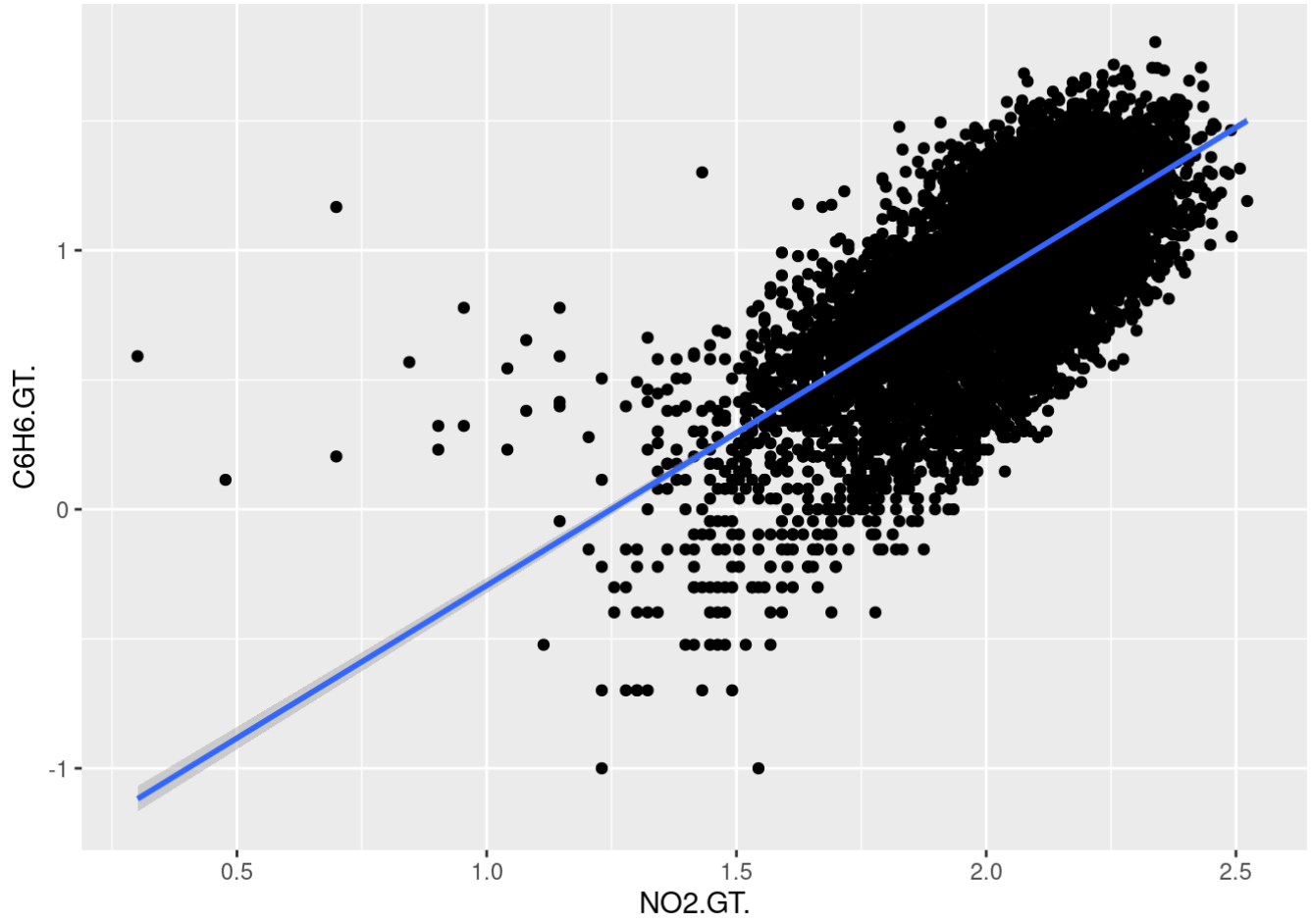
```
## Warning: Removed 366 rows containing missing values (geom_point).
```




```
airqual_norm %>%
  ggplot(aes(x = NO2.GT., y = C6H6.GT.)) +
  geom_point() +
  geom_smooth(method = 'lm')
```

```
## Warning: Removed 1964 rows containing non-finite values (stat_smooth).
```

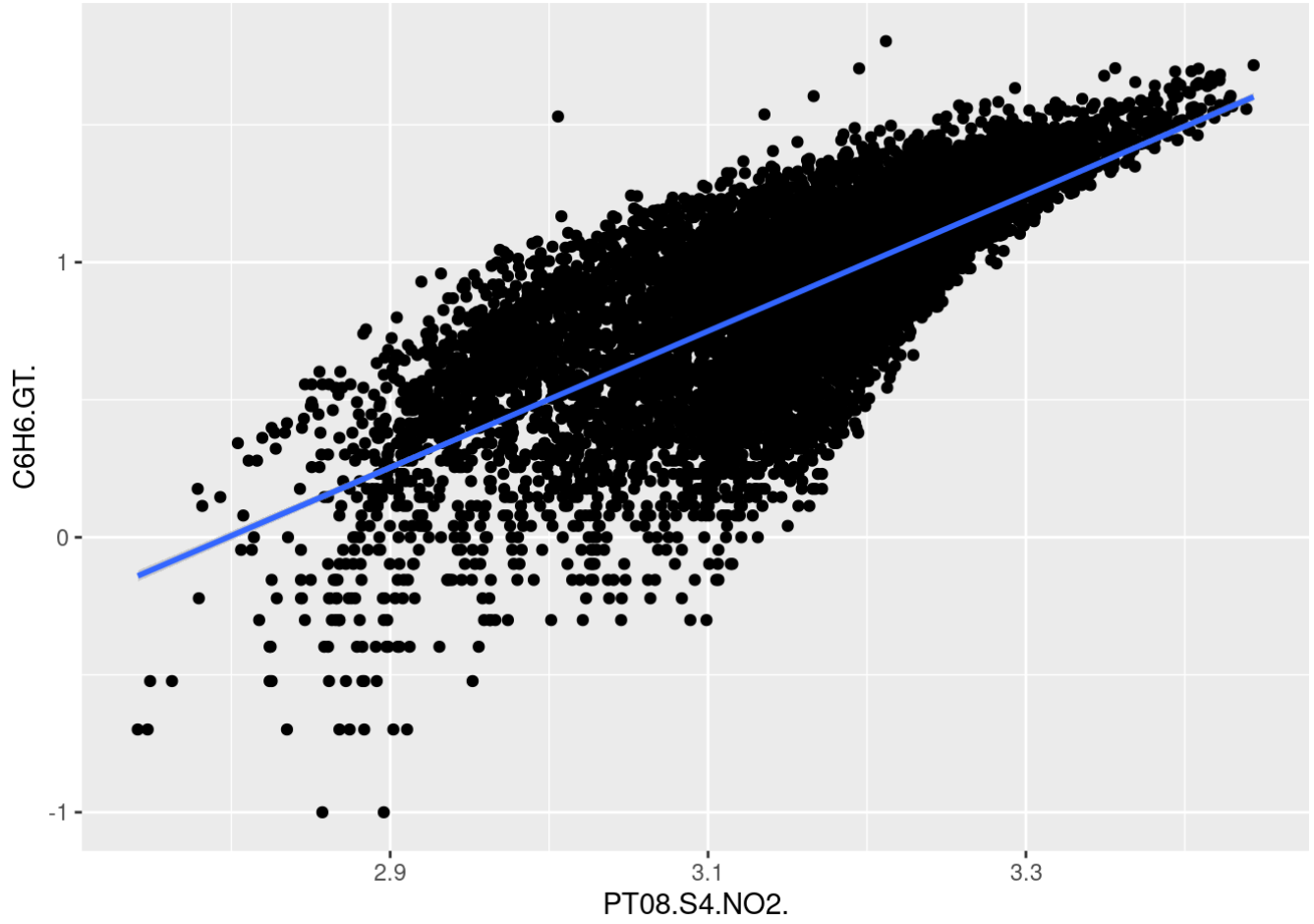
```
## Warning: Removed 1964 rows containing missing values (geom_point).
```




```
airqual_norm %>%
  ggplot(aes(x = PT08.S4.NO2., y = C6H6.GT.)) +
  geom_point() +
  geom_smooth(method = 'lm')
```

```
## Warning: Removed 366 rows containing non-finite values (stat_smooth).
```

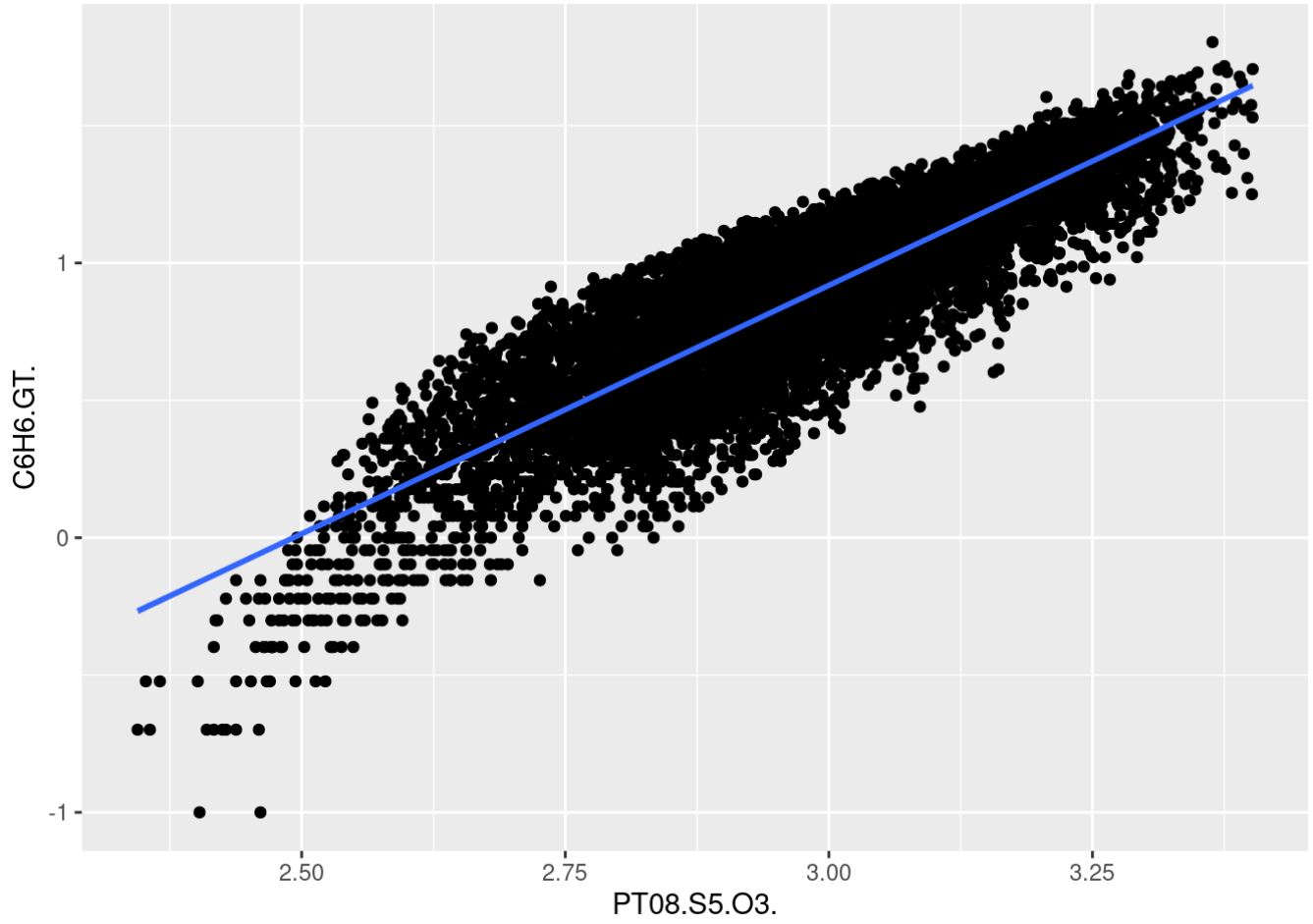
```
## Warning: Removed 366 rows containing missing values (geom_point).
```




```
airqual_norm %>%
  ggplot(aes(x = PT08.S5.O3., y = C6H6.GT.)) +
  geom_point() +
  geom_smooth(method = 'lm')
```

```
## Warning: Removed 366 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 366 rows containing missing values (geom_point).
```




```
set.seed(0)
```

Creating train/test data

```
aq_sample <- sample.int(n = nrow(airqual_norm), size = floor(.75*nrow(airqual_norm)))
train_air_qual <- airqual_norm[aq_sample, ]
test_air_qual <- airqual_norm[-aq_sample, ]
```

Building models

```
l_m_CO <- lm(data=train_air_qual, C6H6.GT.~CO.GT.)
CO_sum <- summary(l_m_CO)
R_CO <- round(CO_sum$r.squared, 4)
p_CO <- round(CO_sum$coefficients[2, 4], 4)
title_CO <- paste('R^2 =', as.character(R_CO), 'p-value =', as.character(p_CO))
```

```
l_m_NO <- lm(data=train_air_qual, C6H6.GT.~NOx.GT.)
NO_sum <- summary(l_m_NO)
R_NO <- round(NO_sum$r.squared, 4)
p_NO <- round(NO_sum$coefficients[2, 4], 4)
title_NO <- paste('R^2 =', as.character(R_NO), 'p-value =', as.character(p_NO))
```

```
l_m_NO2 <- lm(data=train_air_qual, C6H6.GT.~NO2.GT.)
NO2_sum <-summary(l_m_NO2)
R_NO2 <- round(NO2_sum$r.squared, 4)
p_NO2 <- round(NO2_sum$coefficients[2, 4], 4)
title_NO2 <- paste('R^2 =', as.character(R_NO2), 'p-value =', as.character(p_NO2))
```

Prediction of the test data

```
pred_CO <- predict(l_m_CO, newdata = test_air_qual)
test_air_qual$pred_by_CO <- pred_CO

pred_NO <- predict(l_m_NO, newdata = test_air_qual)
test_air_qual$pred_by_NO <- pred_NO

pred_NO2 <- predict(l_m_NO2, newdata = test_air_qual)
test_air_qual$pred_by_NO2 <- pred_NO2
```

```
test <- test_air_qual %>%
  select(C6H6.GT., pred_by_CO, CO.GT., pred_by_NO, NOx.GT., pred_by_NO2, NO2.GT.) %>%
  drop_na()

head(test)
```

```
##     C6H6.GT. pred_by_CO     CO.GT. pred_by_NO  NOx.GT. pred_by_NO2  NO2.GT.
## 1  1.0755470  1.0731941  0.4149733  0.8822814 2.220108   0.9444223 2.053078
## 13 0.2041200  0.5127170 -0.1549020  0.4230161 1.531479   0.5120595 1.681241
## 17 0.7993405  0.8917128  0.2304489  0.7683102 2.049218   0.8725020 1.991226
## 20 0.8633229  0.9392208  0.2787536  0.8450967 2.164353   0.9399335 2.049218
## 23 0.9190781  1.0018399  0.3424227  0.9259312 2.285557   1.0190641 2.117271
## 25 1.3180633  1.3350707  0.6812412  1.0347397 2.448706   1.0908139 2.178977
```

Building plots

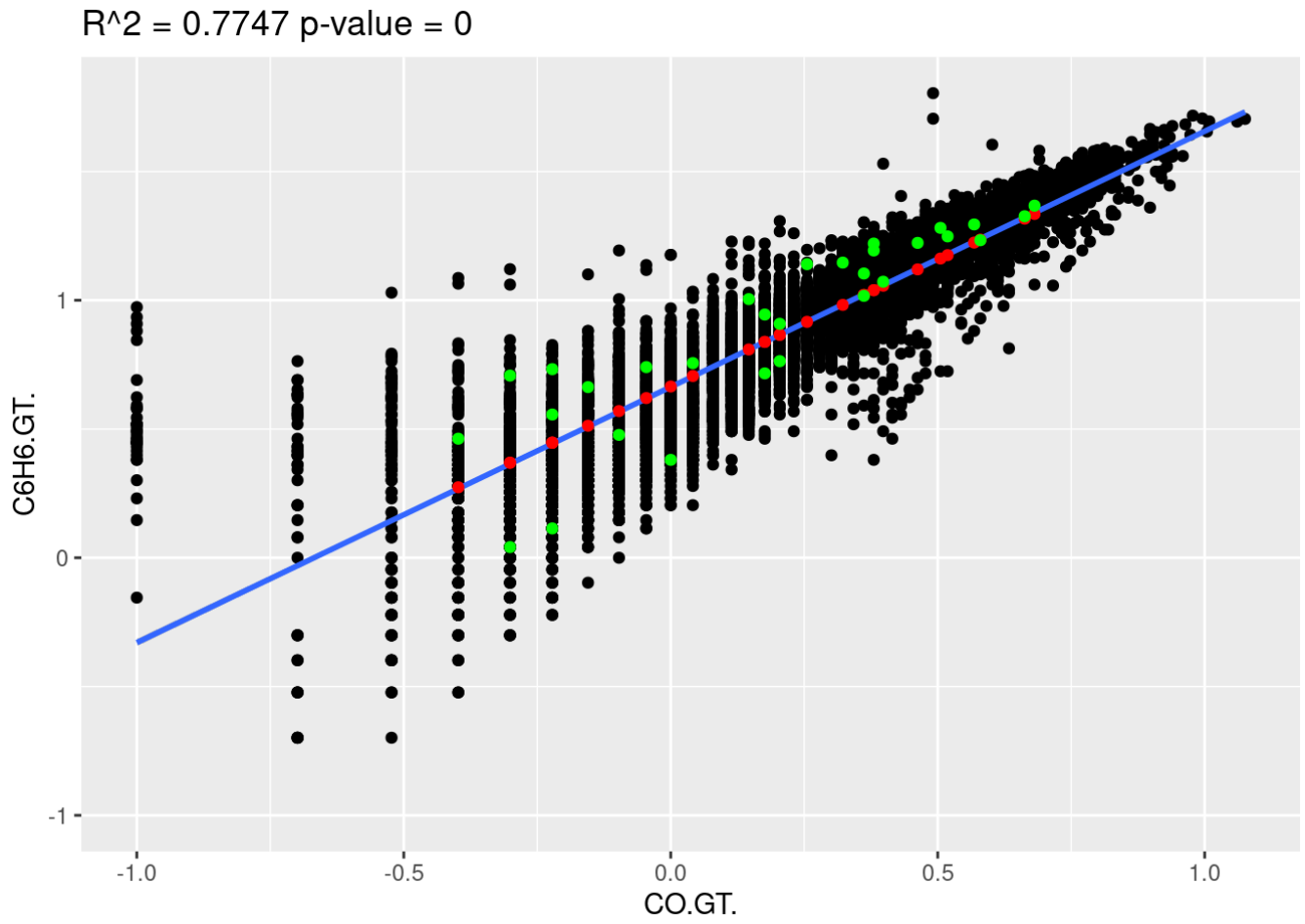
To plot graph with predict and test data I only chose 30 random values. (If plot all predicted values they cover over linear regression line)

```
test_for_graph <- test[sample(rownames(test), size = 30), ]
```

```
ggplot() +
  geom_point(data = airqual_norm, aes(x = CO.GT., y = C6H6.GT.)) +
  geom_smooth(data = airqual_norm, method = 'lm', aes(CO.GT., C6H6.GT.)) +
  geom_point(data = test_for_graph, aes(x = CO.GT., y = pred_by_CO), color = 'red') +
  geom_point(data = test_for_graph, aes(x = CO.GT., y = C6H6.GT.), color = 'green') +
  labs(title = title_CO)
```

```
## Warning: Removed 2013 rows containing non-finite values (stat_smooth).
```

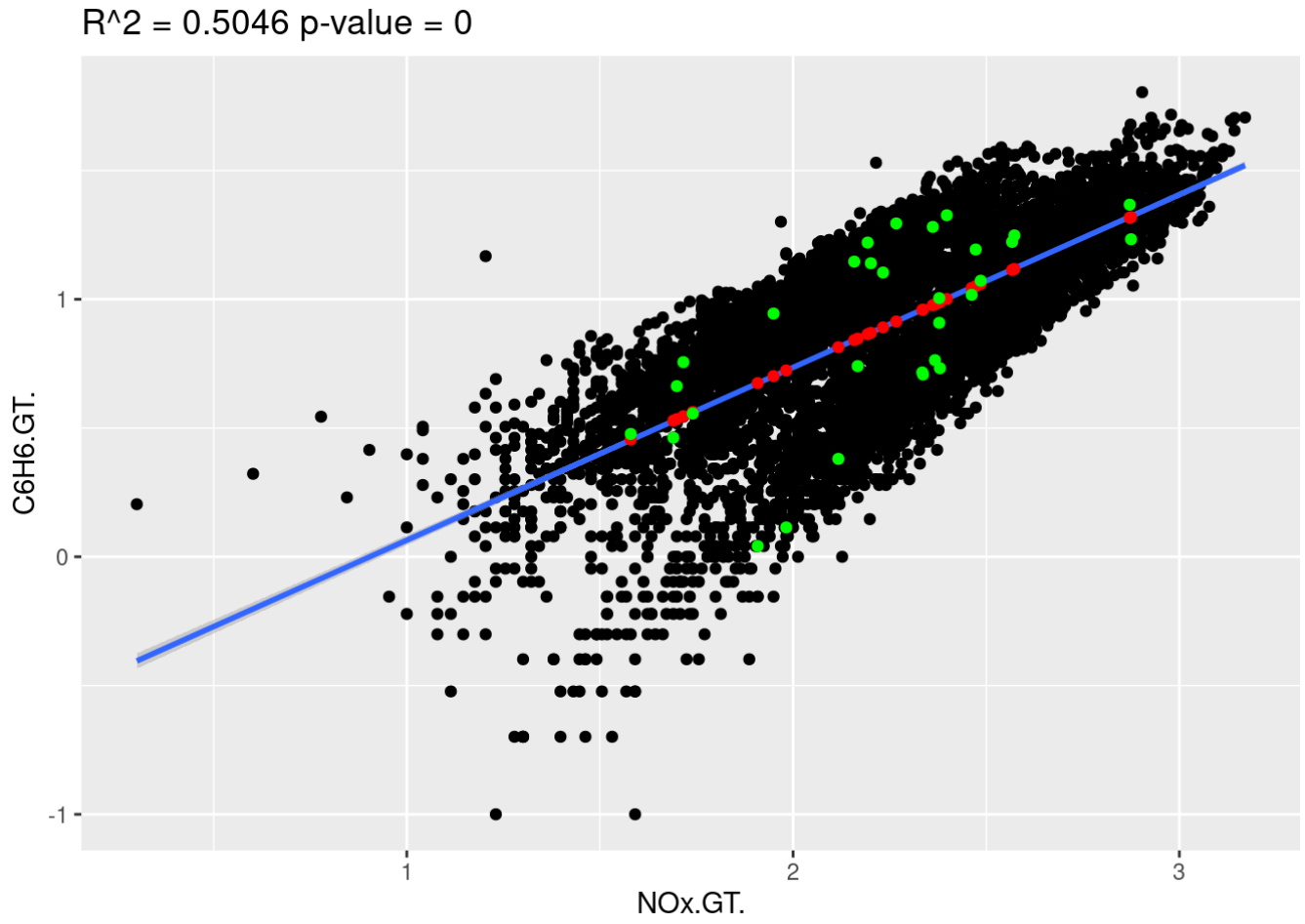
```
## Warning: Removed 2013 rows containing missing values (geom_point).
```




```
ggplot() +
  geom_point(data = airqual_norm, aes(x = NOx.GT., y = C6H6.GT.)) +
  geom_smooth(data = airqual_norm, method = 'lm', aes(NOx.GT., C6H6.GT.)) +
  geom_point(data = test_for_graph, aes(x = NOx.GT., y = pred_by_NO), color = 'red') +
  geom_point(data = test_for_graph, aes(x = NOx.GT., y = C6H6.GT.), color = 'green') +
  labs(title = title_NO)
```

```
## Warning: Removed 1961 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1961 rows containing missing values (geom_point).
```




```
ggplot() +
  geom_point(data = airqual_norm, aes(x = NO2.GT., y = C6H6.GT.)) +
  geom_smooth(data = airqual_norm, method = 'lm', aes(NO2.GT., C6H6.GT.)) +
  geom_point(data = test_for_graph, aes(x = NO2.GT., y = pred_by_NO2), color = 'red') +
  geom_point(data = test_for_graph, aes(x = NO2.GT., y = C6H6.GT.), color = 'green') +
  labs(title = title_NO2)
```

```
## Warning: Removed 1964 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1964 rows containing missing values (geom_point).
```