Homework2_1

Valeriia

31 03 2020

```
library(dplyr)
library(ggplot2)
library(tidyr)
library(ggplot2)
library(reshape2)
library(data.table)
```

First task

Import data and make it long

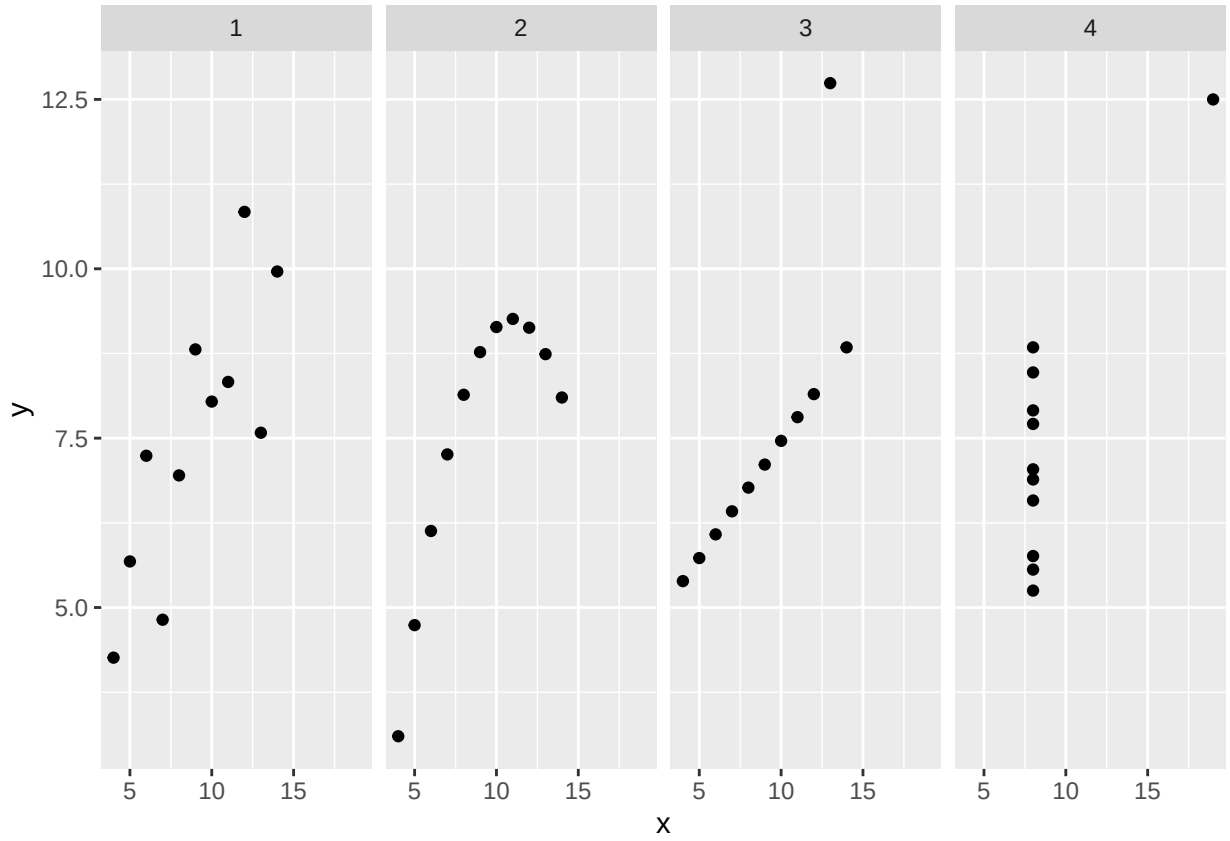
```
ans <- datasets::anscombe
ansl <- data.frame(
    group  = rep(1:4, each = 11),
    x = unlist(ans[,c(1:4)]),
    y = unlist(ans[,c(5:8)])
    )
rownames(ansl) <- NULL
ansl
```

```
##    group  x     y
## 1      1 10  8.04
## 2      1  8  6.95
## 3      1 13  7.58
## 4      1  9  8.81
## 5      1 11  8.33
## 6      1 14  9.96
## 7      1  6  7.24
## 8      1  4  4.26
## 9      1 12 10.84
## 10     1  7  4.82
## 11     1  5  5.68
## 12     2 10  9.14
## 13     2  8  8.14
## 14     2 13  8.74
## 15     2  9  8.77
## 16     2 11  9.26
## 17     2 14  8.10
## 18     2  6  6.13
## 19     2  4  3.10
```

```
## 20     2 12  9.13
## 21     2  7  7.26
## 22     2  5  4.74
## 23     3 10  7.46
## 24     3  8  6.77
## 25     3 13 12.74
## 26     3  9  7.11
## 27     3 11  7.81
## 28     3 14  8.84
## 29     3  6  6.08
## 30     3  4  5.39
## 31     3 12  8.15
## 32     3  7  6.42
## 33     3  5  5.73
## 34     4  8  6.58
## 35     4  8  5.76
## 36     4  8  7.71
## 37     4  8  8.84
## 38     4  8  8.47
## 39     4  8  7.04
## 40     4  8  5.25
## 41     4 19 12.50
## 42     4  8  5.56
## 43     4  8  7.91
## 44     4  8  6.89
```

Build a plot

```
ggplot(ansl, aes(x=x, y=y)) + geom_point()+facet_grid(.~group)
```



Correlation and mean, sd.

```
ansl %>%
  group_by(group) %>%
  summarise_each(funs(mean, sd))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.
```

```
## # A tibble: 4 x 5
##   group x_mean y_mean  x_sd  y_sd
##   <int>  <dbl>  <dbl> <dbl> <dbl>
## 1     1      9   7.50  3.32  2.03
## 2     2      9   7.50  3.32  2.03
## 3     3      9   7.5   3.32  2.03
## 4     4      9   7.50  3.32  2.03
```

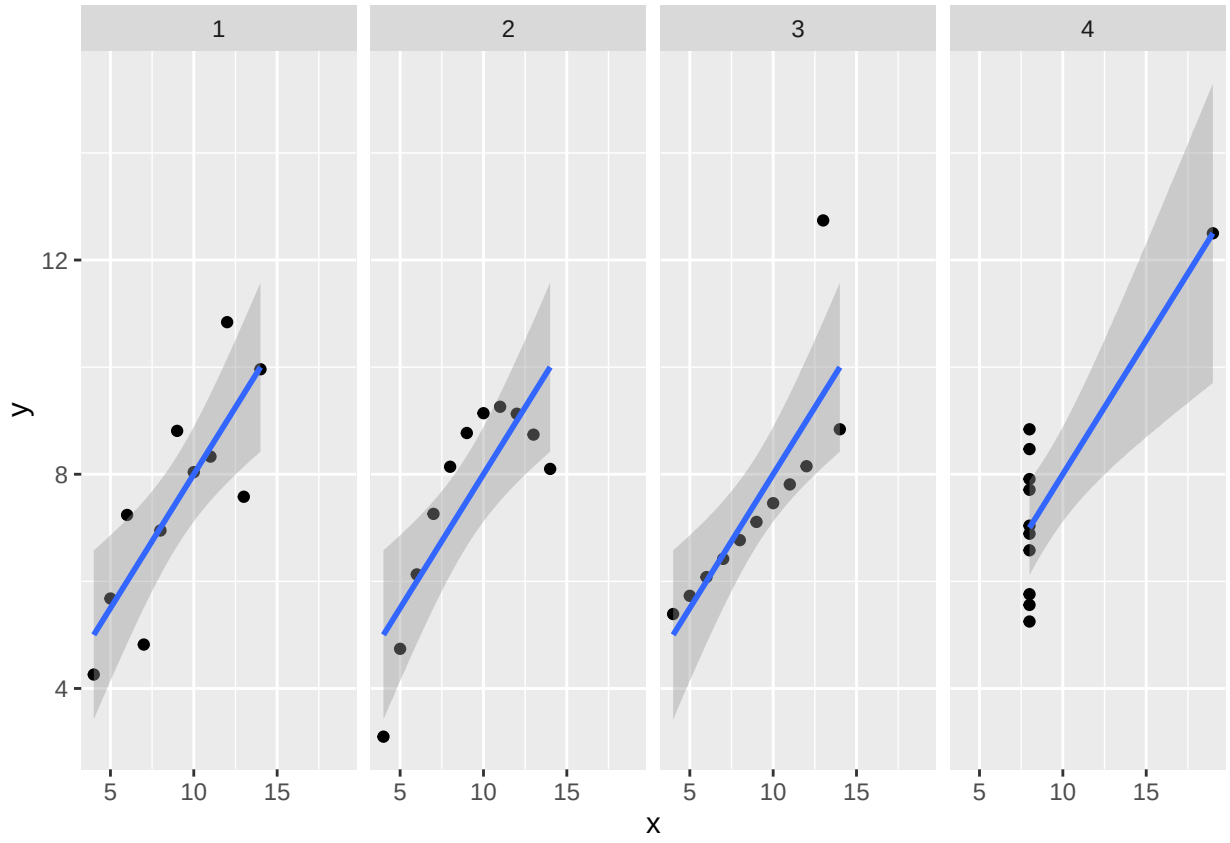
```r
ansl %>%
  group_by(group) %>%
  summarise_at(vars(-y),funs(cor_pears=cor(x, y, method = c("pearson")), cor_spear=cor(x, y, method = c
```

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

Plots

```r
ggplot(ansl, aes(x=x, y=y)) + geom_point()+facet_grid(.~group)+geom_smooth(method = lm)
```

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



Second task

Import data

```
air <- read.csv("Air.csv", sep=";", dec=",")
air <- air[,-c(16, 17)]
air <- air[rowSums(is.na(air[,13:15]))==0,]
```

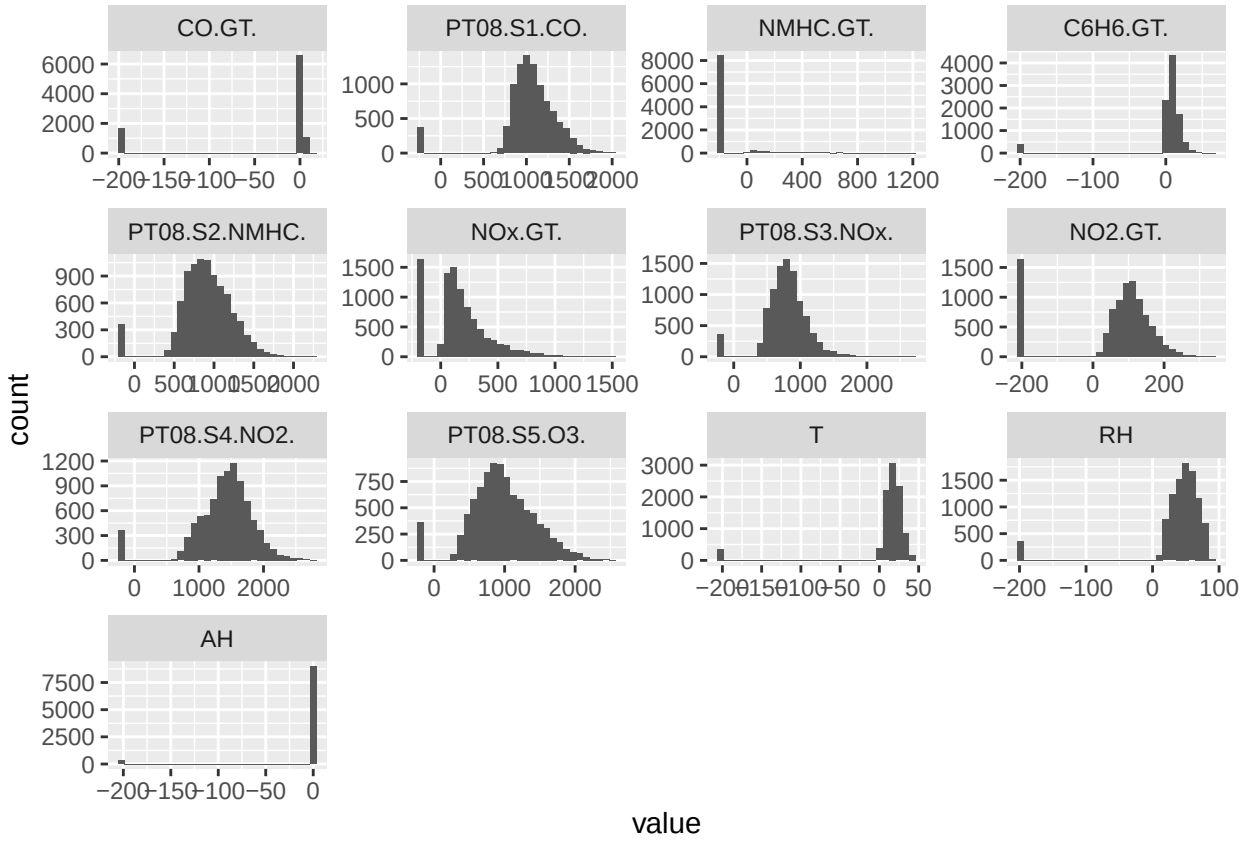
Make it long and check data. We can see outliers in -200 - replace it with NA drop it. Why drop? Because it probably means NA, also we can see it in the several columns in the same row, so this data will make our life harder in regression. Also we can notice that not all columns have normal distribution

```
air_long <- melt(air)
```

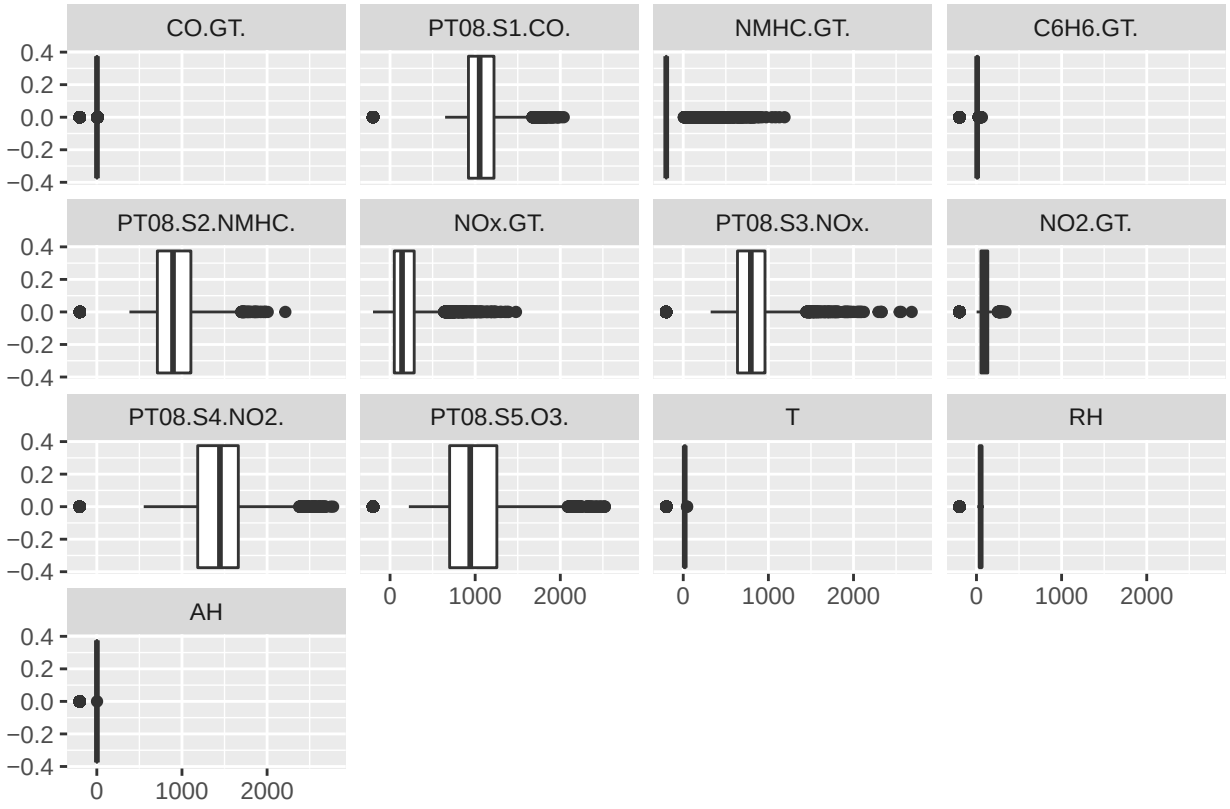
```
## Warning in melt(air): The melt generic in data.table has been passed a
## data.frame and will attempt to redirect to the relevant reshape2 method;
## please note that reshape2 is deprecated, and this redirection is now
## deprecated as well. To continue using melt methods from reshape2 while both
## libraries are attached, e.g. melt.list, you can prepend the namespace like
## reshape2::melt(air). In the next version, this warning will become an error.

## Using Date, Time as id variables
```

```
air_long <- air_long[,-c(1,2)]
ggplot(air_long, aes(value)) +
  geom_histogram() +
  facet_wrap(~variable, scales = "free")
```

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



```
ggplot(air_long, aes(value)) +
  geom_boxplot() +
  facet_wrap(~variable)
```

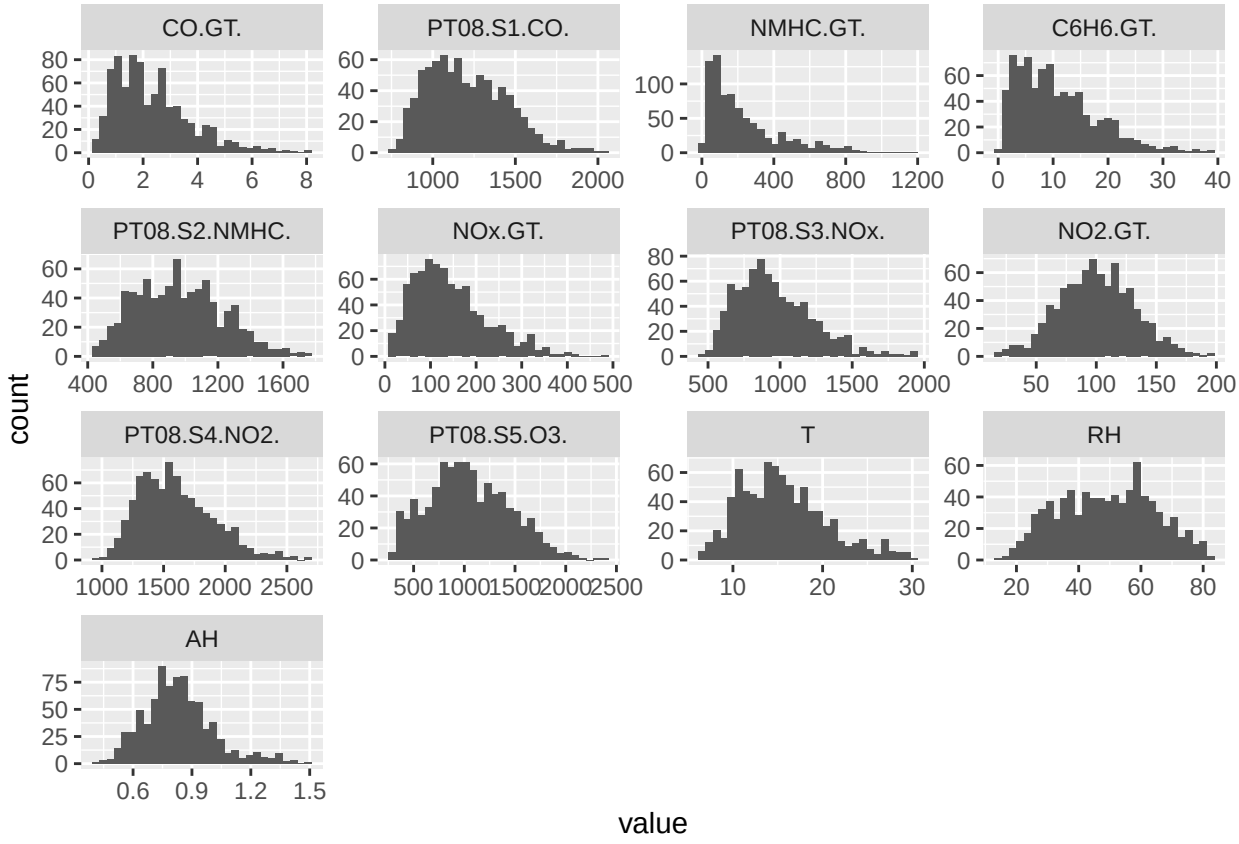



```
for (i in 3:ncol(air)) {
  air[,i][which(air[,i] == -200)] <- NA
}

air <- drop_na(air)

air_long <- melt(air)
```

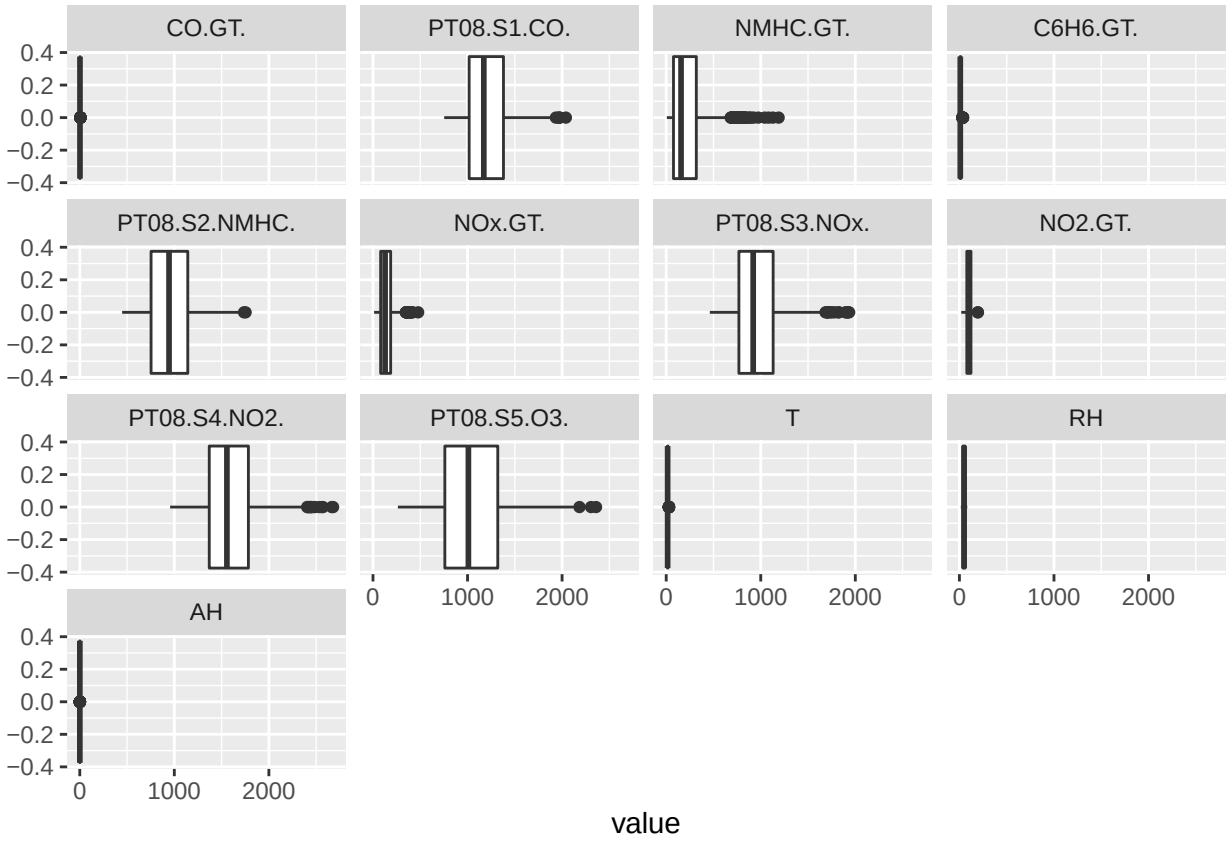
```
## Warning in melt(air): The melt generic in data.table has been passed a
## data.frame and will attempt to redirect to the relevant reshape2 method;
## please note that reshape2 is deprecated, and this redirection is now
## deprecated as well. To continue using melt methods from reshape2 while both
## libraries are attached, e.g. melt.list, you can prepend the namespace like
## reshape2::melt(air). In the next version, this warning will become an error.

## Using Date, Time as id variables
```

```
air_long <- air_long[,-c(1,2)]
ggplot(air_long, aes(value)) +
  geom_histogram() +
  facet_wrap(~variable, scales = "free")
```

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

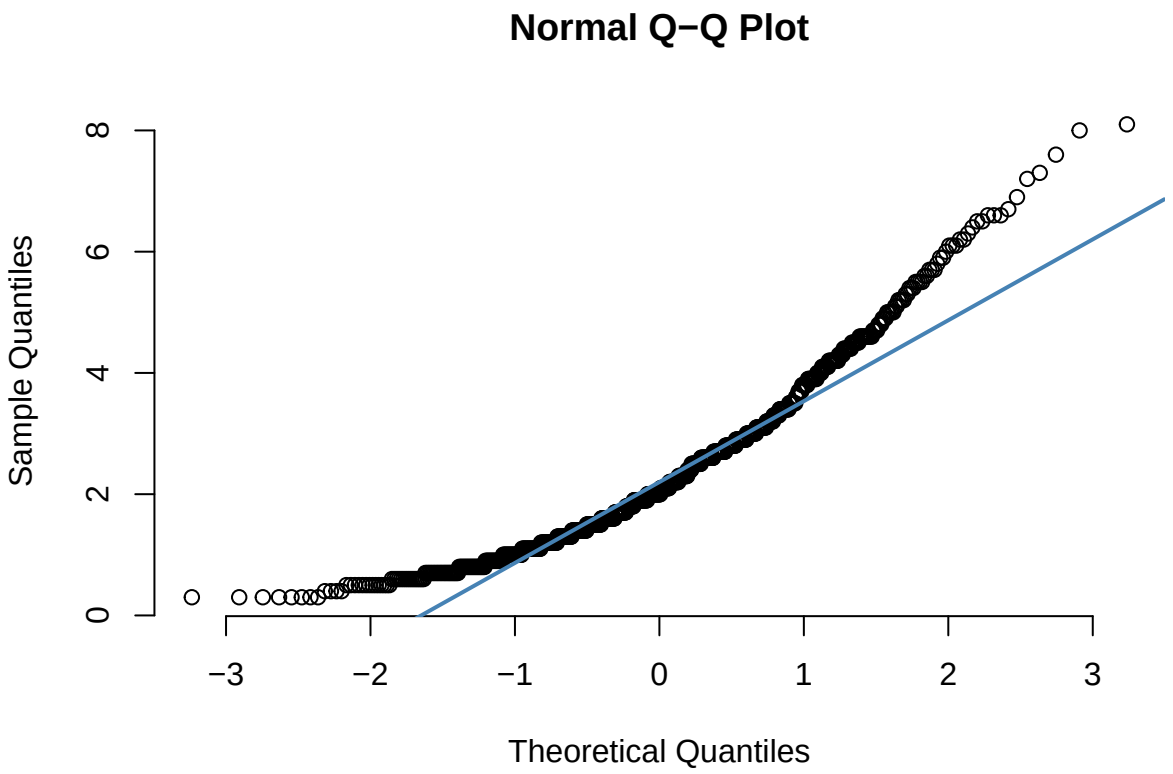



```
ggplot(air_long, aes(value)) +
  geom_boxplot() +
  facet_wrap(~variable)
```

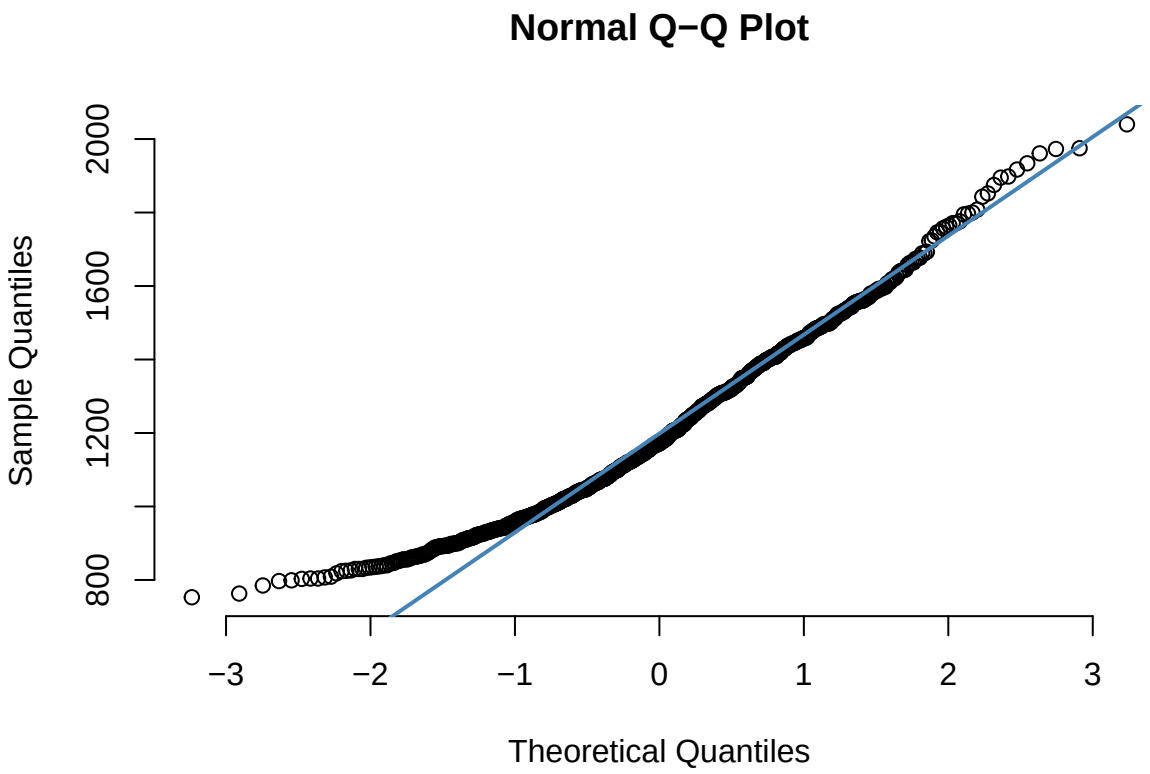


Check our data. Quantile - the proportion of cases that are less than certain values. If the requirements of "normality" are here - the data should lie on a diagonal line. We can see that several columns do not have normal distribution.

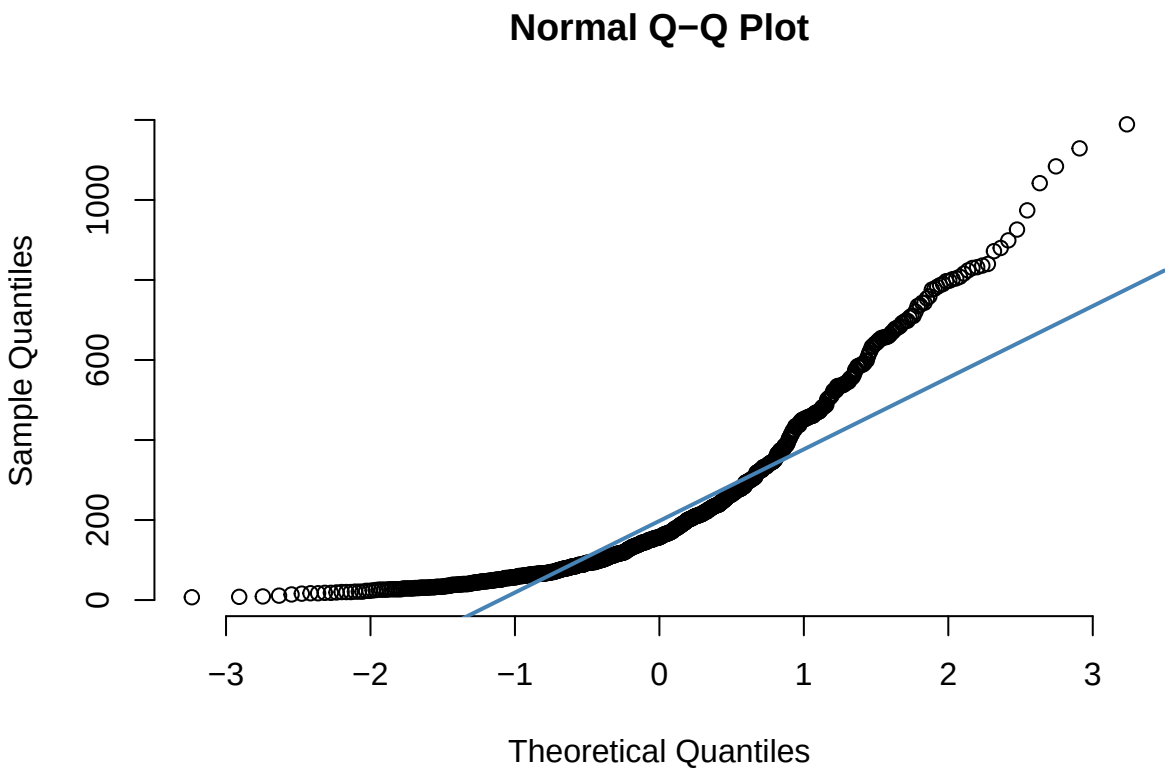
```
#CO.GT
qqnorm(air$CO.GT., pch = 1, frame = FALSE)
qqline(air$CO.GT., col = "steelblue", lwd = 2)
```



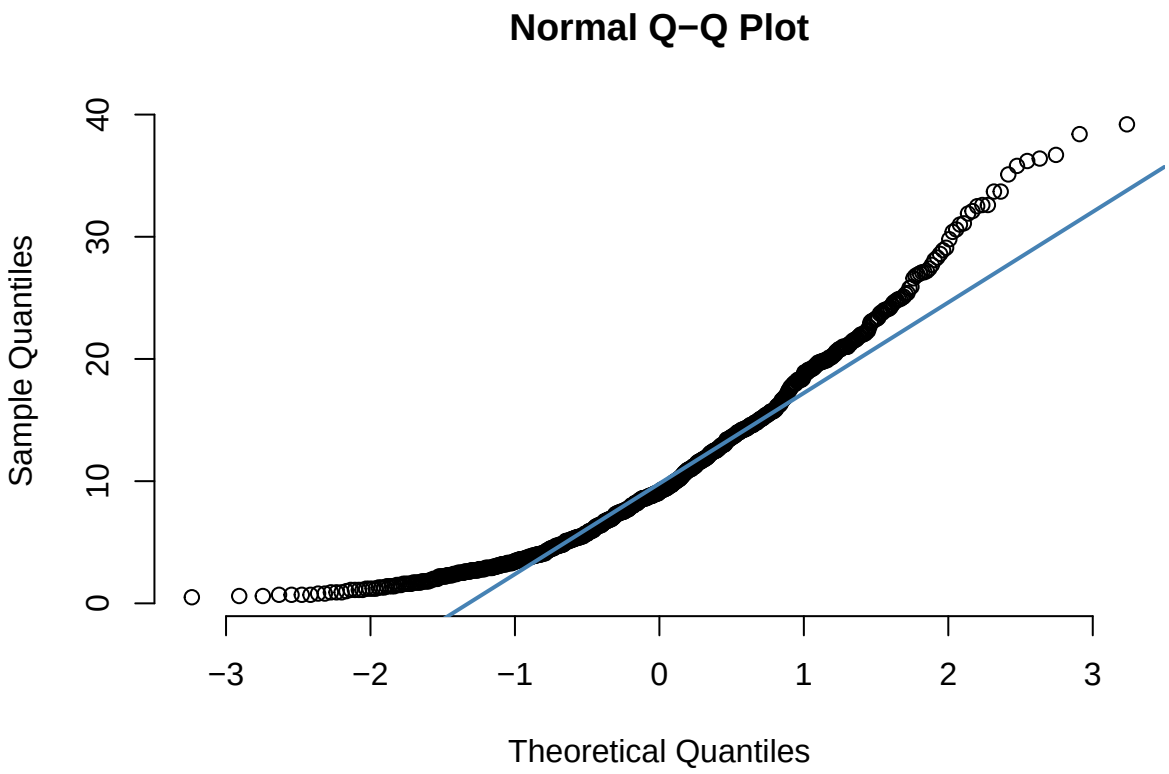

```r
#PT08.S1.CO
qqnorm(air$PT08.S1.CO., pch = 1, frame = FALSE)
qqline(air$PT08.S1.CO., col = "steelblue", lwd = 2)
```



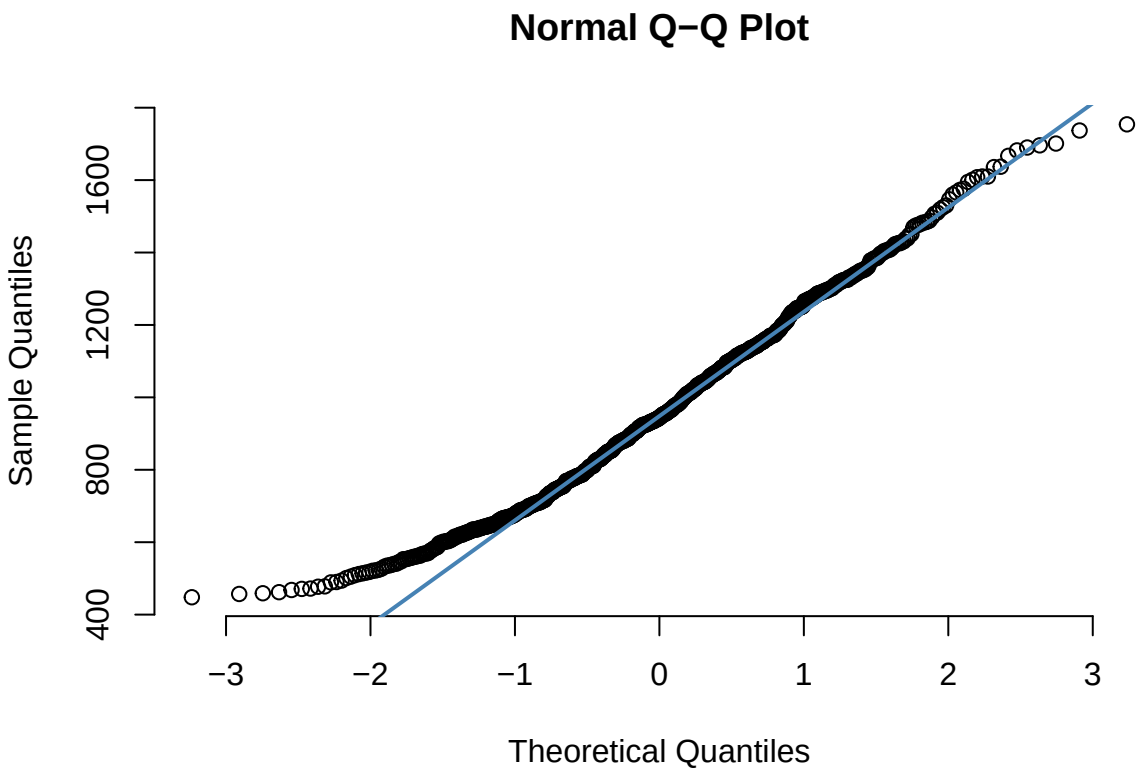

```r
#NMHC.GT
qqnorm(air$NMHC.GT., pch = 1, frame = FALSE)
qqline(air$NMHC.GT., col = "steelblue", lwd = 2)
```



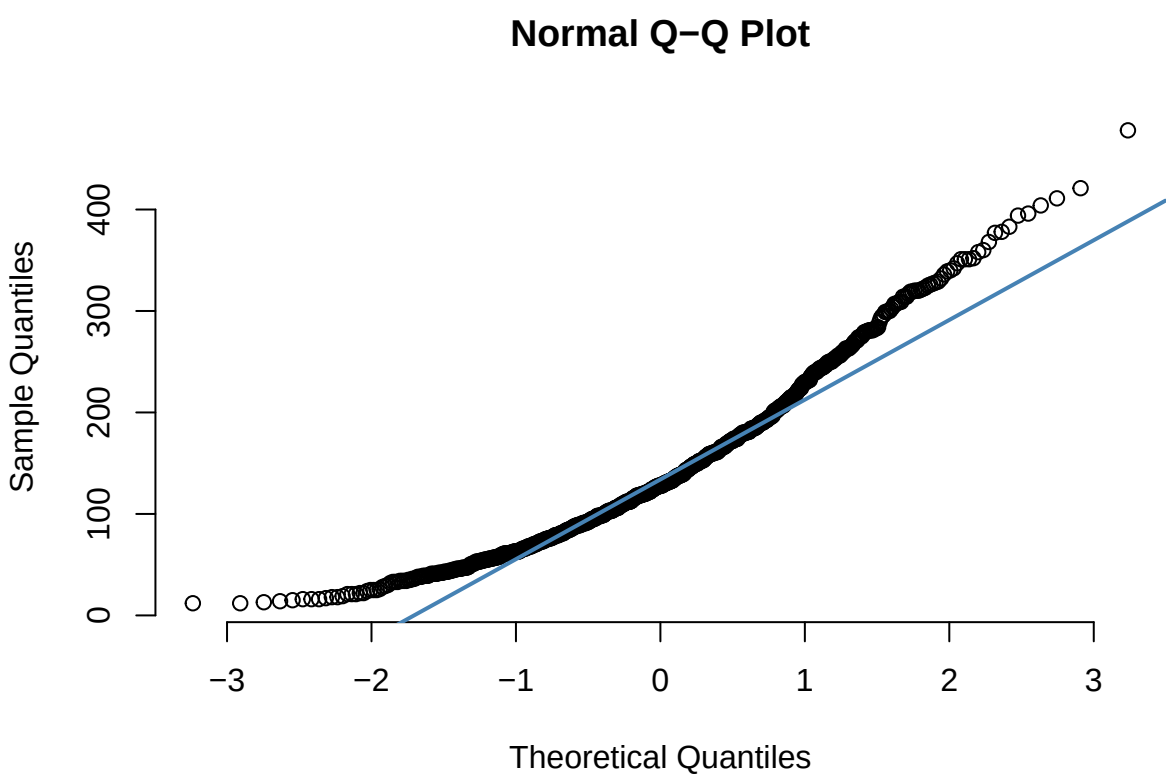

```r
#C6H6.GT
qqnorm(air$C6H6.GT., pch = 1, frame = FALSE)
qqline(air$C6H6.GT., col = "steelblue", lwd = 2)
```



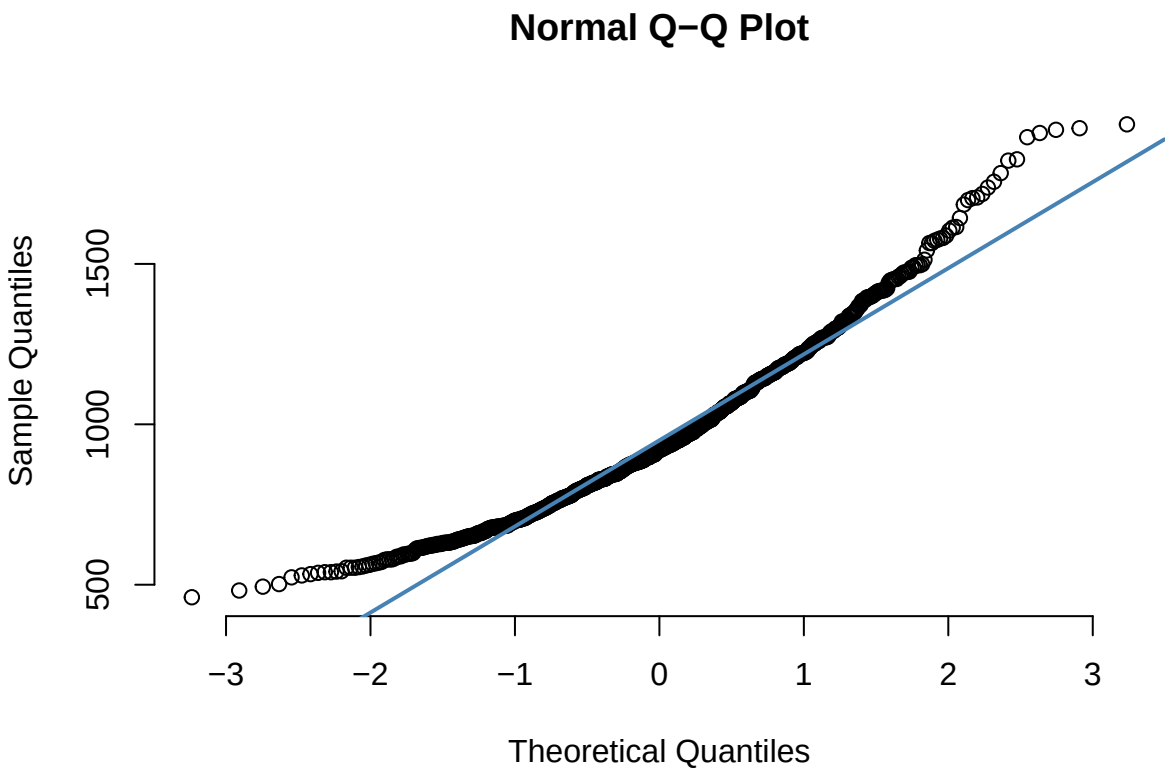

```
#PT08.S2.NMHC
qqnorm(air$PT08.S2.NMHC., pch = 1, frame = FALSE)
qqline(air$PT08.S2.NMHC., col = "steelblue", lwd = 2)
```



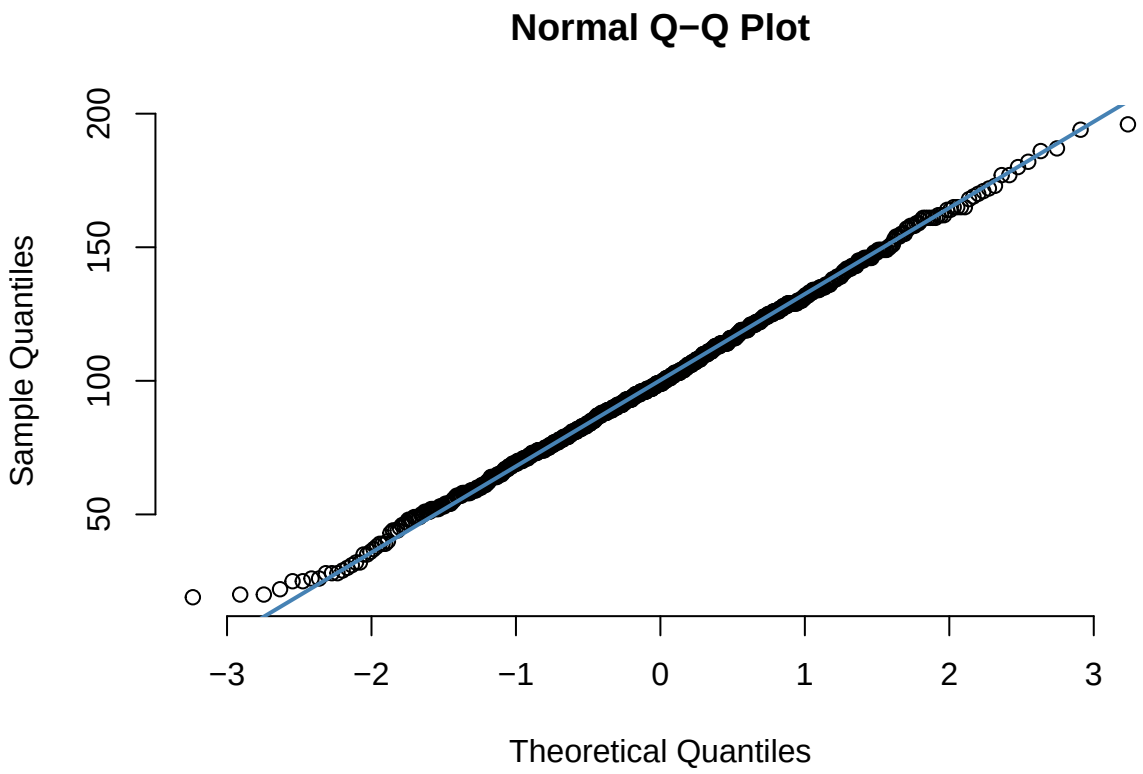

```r
#NOx.GT
qqnorm(air$NOx.GT., pch = 1, frame = FALSE)
qqline(air$NOx.GT., col = "steelblue", lwd = 2)
```



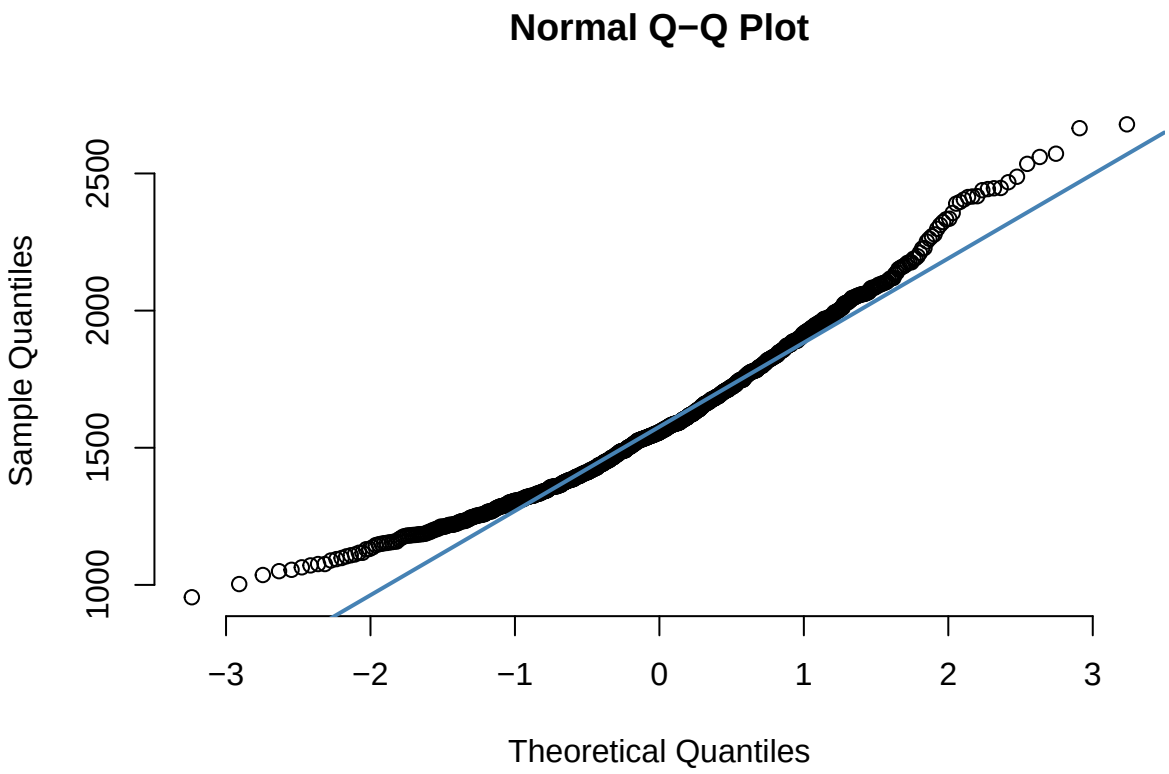

```r
#PT08.S3.NOx
qqnorm(air$PT08.S3.NOx., pch = 1, frame = FALSE)
qqline(air$PT08.S3.NOx., col = "steelblue", lwd = 2)
```



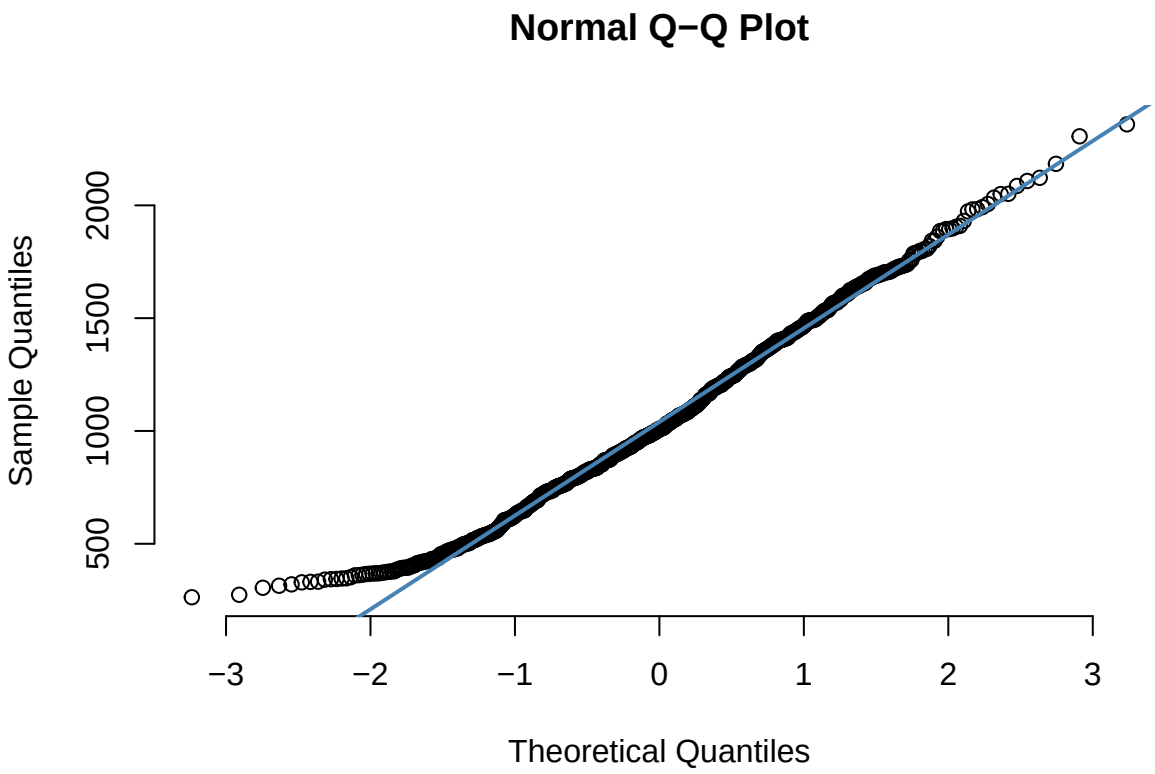

```
#NO2.GT
qqnorm(air$NO2.GT., pch = 1, frame = FALSE)
qqline(air$NO2.GT., col = "steelblue", lwd = 2)
```



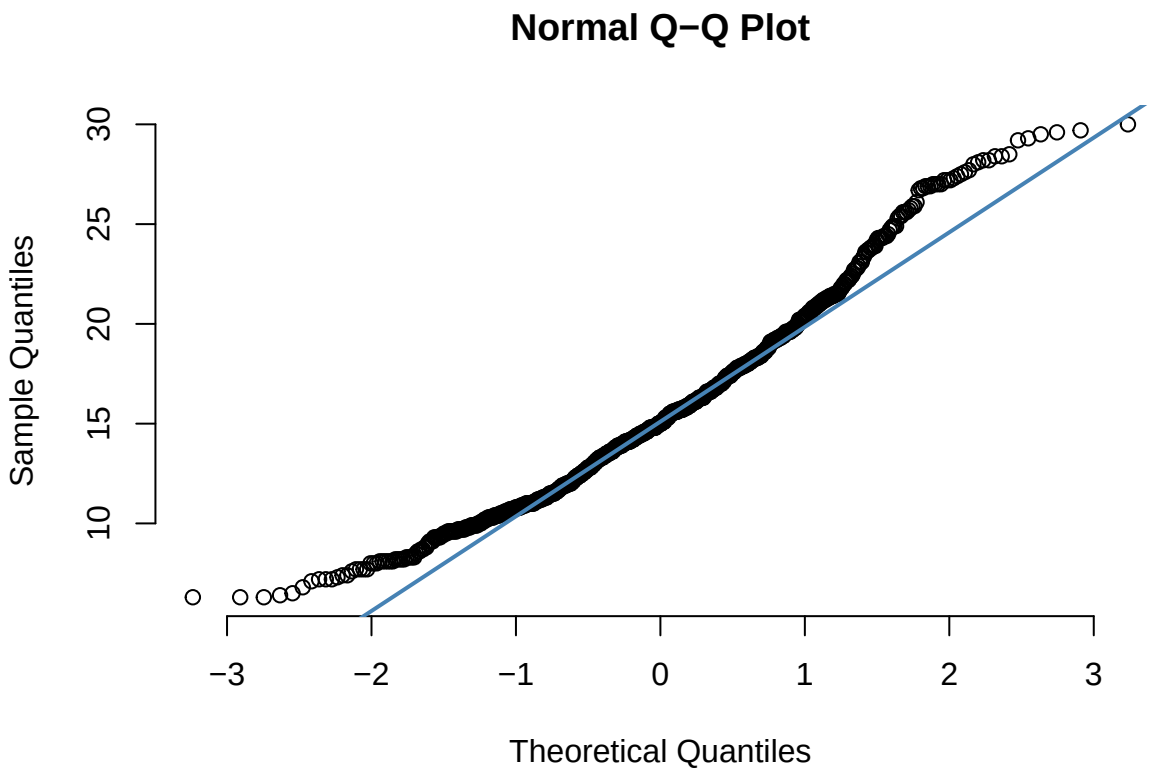

```r
#PT08.S4.NO2
qqnorm(air$PT08.S4.NO2., pch = 1, frame = FALSE)
qqline(air$PT08.S4.NO2., col = "steelblue", lwd = 2)
```



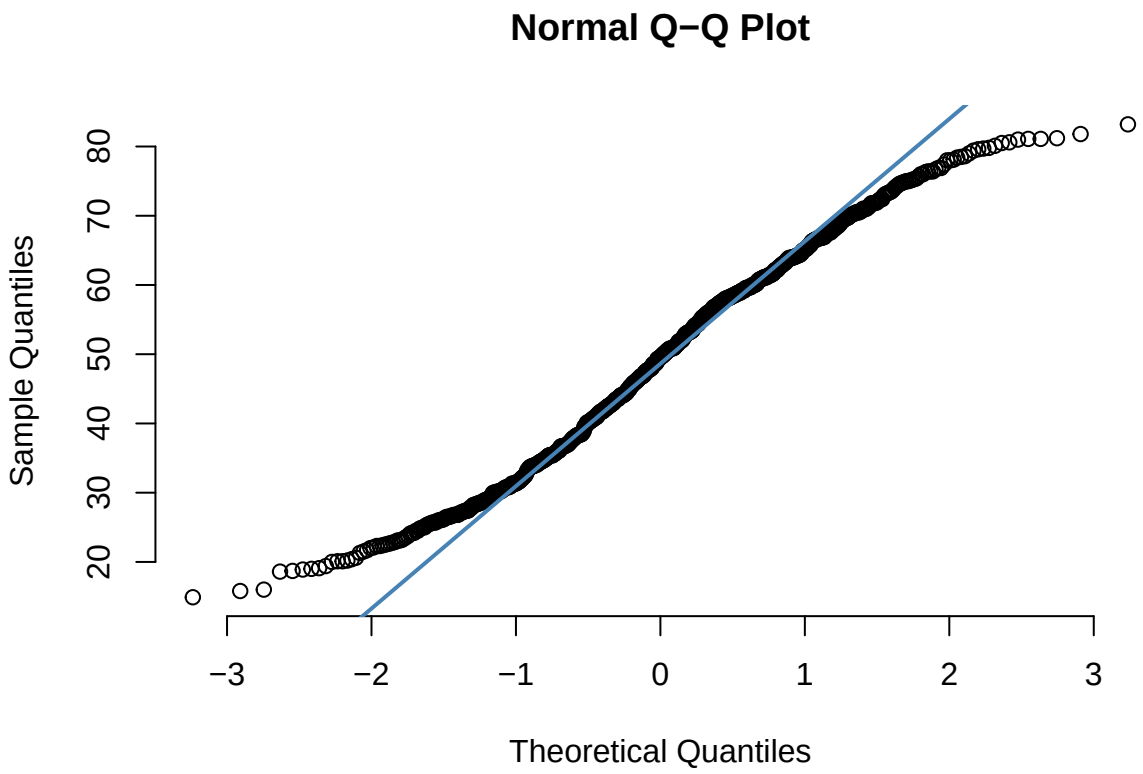

```r
#PT08.S5.O3
qqnorm(air$PT08.S5.O3., pch = 1, frame = FALSE)
qqline(air$PT08.S5.O3., col = "steelblue", lwd = 2)
```



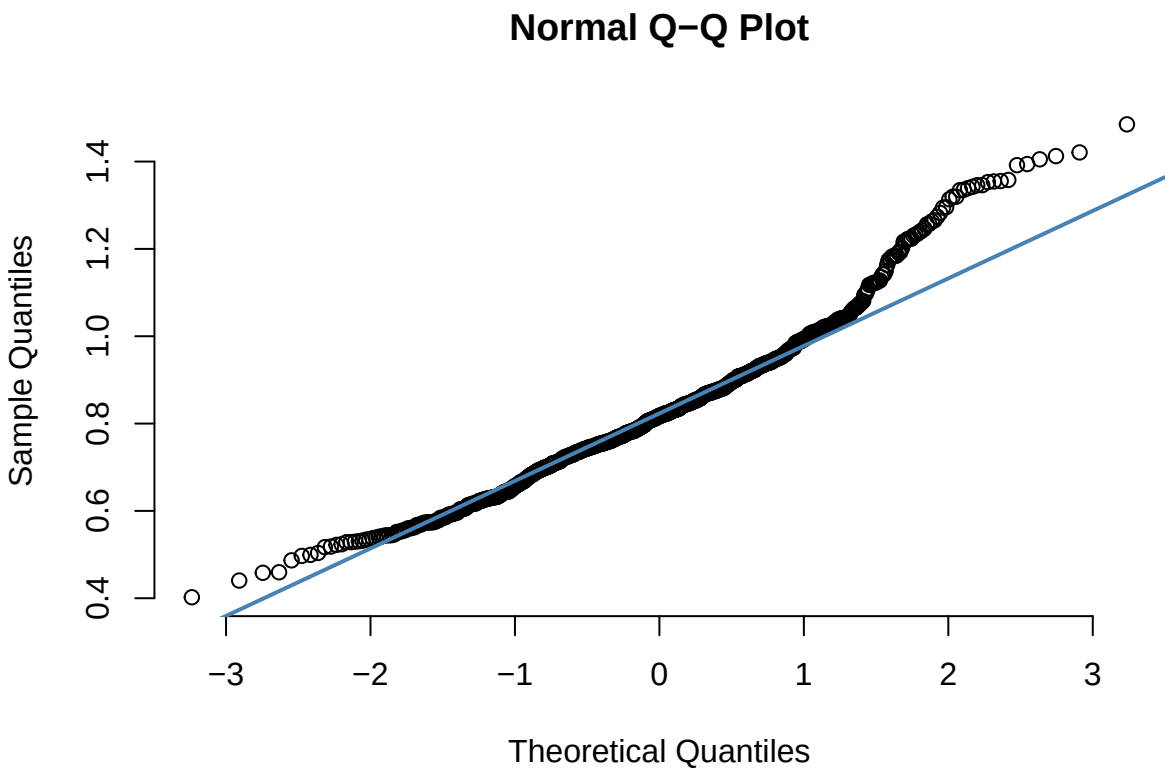

```
#T
qqnorm(air$T, pch = 1, frame = FALSE)
qqline(air$T, col = "steelblue", lwd = 2)
```




```r
#RH
qqnorm(air$RH, pch = 1, frame = FALSE)
qqline(air$RH, col = "steelblue", lwd = 2)
```

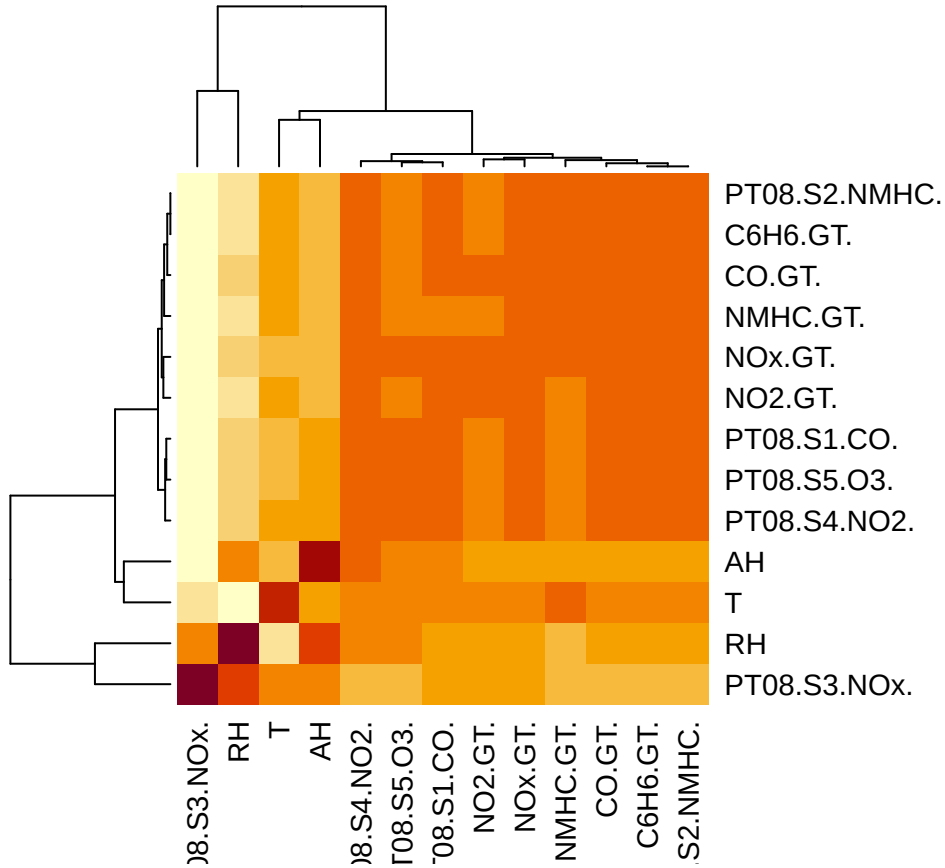



```r
#AH
qqnorm(air$AH, pch = 1, frame = FALSE)
qqline(air$AH, col = "steelblue", lwd = 2)
```



Cross correlation. Red - high correlation, white - low.

```
ccs <- as.matrix(air[,c(3:15)])
heatmap(cor(ccs,use = "complete.obs", method = "spearman"))
```




```
cor(ccs, use = "complete.obs", method = "spearman")
```

```
##                  CO.GT. PT08.S1.CO.   NMHC.GT.   C6H6.GT. PT08.S2.NMHC.
## CO.GT.         1.0000000  0.93978597  0.9342883  0.9766091     0.9766404
## PT08.S1.CO.    0.9397860  1.00000000  0.8386699  0.9348855     0.9349883
## NMHC.GT.       0.9342883  0.83866992  1.0000000  0.9454609     0.9454232
## C6H6.GT.       0.9766091  0.93488550  0.9454609  1.0000000     0.9999832
## PT08.S2.NMHC.  0.9766404  0.93498826  0.9454232  0.9999832     1.0000000
## NOx.GT.        0.9615010  0.92622189  0.8805828  0.9431609     0.9432691
## PT08.S3.NOx.  -0.9235819 -0.89118253 -0.9461075 -0.9526035    -0.9526213
## NO2.GT.        0.9172671  0.88351864  0.8354841  0.8989692     0.8990827
## PT08.S4.NO2.   0.9330848  0.94639571  0.8924281  0.9576110     0.9576513
## PT08.S5.O3.    0.8828508  0.93386561  0.8017011  0.9002137     0.9002616
## T              0.3910139  0.34464297  0.4784004  0.4674760     0.4671404
## RH            -0.1632868 -0.05029757 -0.2410037 -0.2154793    -0.2151061
## AH             0.2947413  0.43470088  0.3163970  0.3321598     0.3323398
##                   NOx.GT. PT08.S3.NOx.    NO2.GT. PT08.S4.NO2.  PT08.S5.O3.
## CO.GT.         0.96150101  -0.92358193  0.9172671   0.93308480  0.882850756
## PT08.S1.CO.    0.92622189  -0.89118253  0.8835186   0.94639571  0.933865605
## NMHC.GT.       0.88058277  -0.94610750  0.8354841   0.89242814  0.801701058
## C6H6.GT.       0.94316091  -0.95260345  0.8989692   0.95761101  0.900213719
## PT08.S2.NMHC.  0.94326915  -0.95262132  0.8990827   0.95765129  0.900261575
## NOx.GT.        1.00000000  -0.89048355  0.9088018   0.91767242  0.891077771
## PT08.S3.NOx.  -0.89048355   1.00000000 -0.8205646  -0.95473807 -0.897821592
## NO2.GT.        0.90880183  -0.82056461  1.0000000   0.84413499  0.835499186
```

```
## PT08.S4.NO2.   0.91767242  -0.95473807  0.8441350   1.00000000  0.926023708
## PT08.S5.O3.    0.89107777  -0.89782159  0.8354992   0.92602371  1.000000000
## T              0.30603518  -0.45075496  0.3991110   0.38412086  0.321566434
## RH            -0.08554217   0.09876639 -0.2250338  -0.01714565 -0.004638448
## AH             0.29736842  -0.50185884  0.2386725   0.53597752  0.479249063
##                        T           RH         AH
## CO.GT.         0.3910139 -0.163286796  0.2947413
## PT08.S1.CO.    0.3446430 -0.050297575  0.4347009
## NMHC.GT.       0.4784004 -0.241003737  0.3163970
## C6H6.GT.       0.4674760 -0.215479340  0.3321598
## PT08.S2.NMHC.  0.4671404 -0.215106064  0.3323398
## NOx.GT.        0.3060352 -0.085542171  0.2973684
## PT08.S3.NOx.  -0.4507550  0.098766391 -0.5018588
## NO2.GT.        0.3991110 -0.225033805  0.2386725
## PT08.S4.NO2.   0.3841209 -0.017145651  0.5359775
## PT08.S5.O3.    0.3215664 -0.004638448  0.4792491
## T              1.0000000 -0.778027606  0.1490804
## RH            -0.7780276  1.000000000  0.4512274
## AH             0.1490804  0.451227370  1.0000000
```

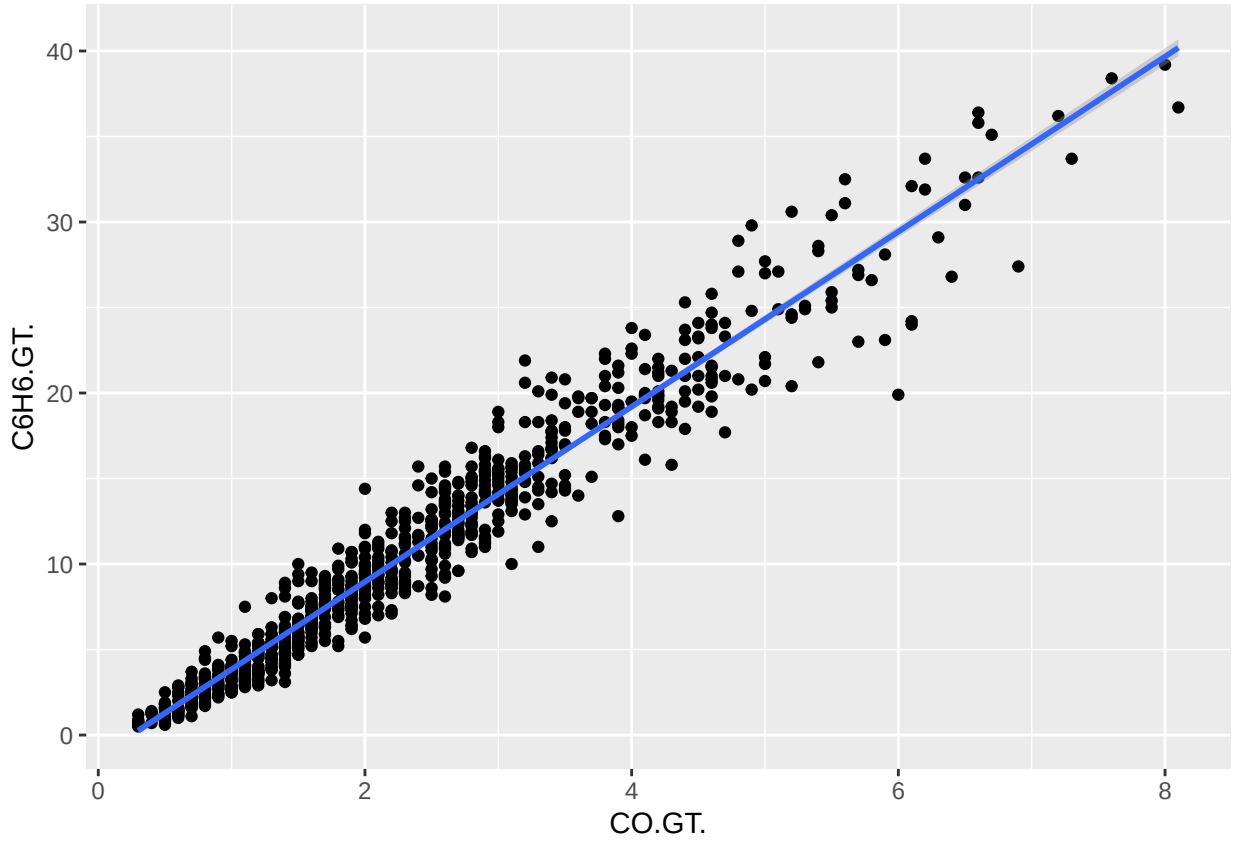
Response C6H6.GT.

```
air%>%
   lm(data = ., C6H6.GT. ~ CO.GT.)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ CO.GT., data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5375 -0.9541 -0.1064  0.8293  6.7959
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.27699    0.11672  -10.94   <2e-16 ***
## CO.GT.       5.11908    0.04255  120.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.724 on 825 degrees of freedom
## Multiple R-squared:  0.9461, Adjusted R-squared:  0.946
## F-statistic: 1.447e+04 on 1 and 825 DF,  p-value: < 2.2e-16
```

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

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



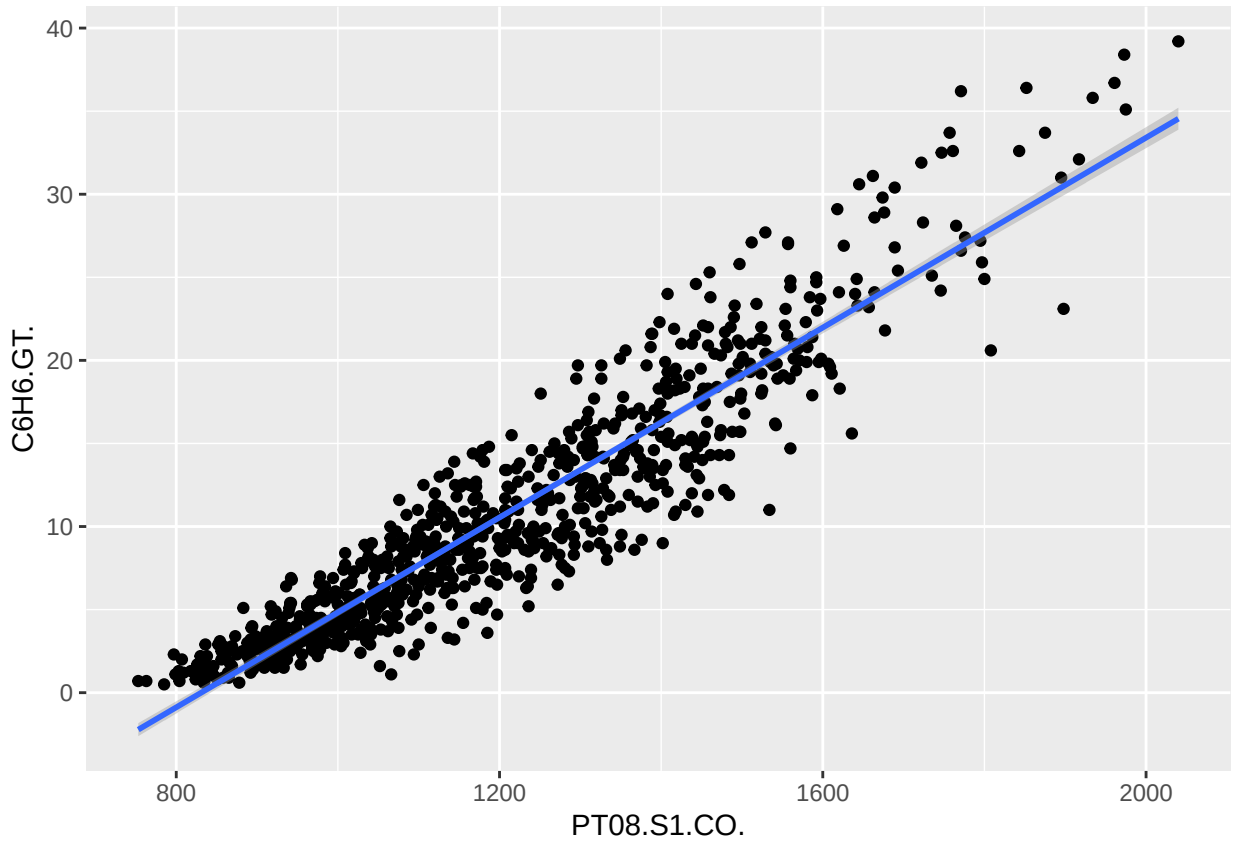

```
air%>%
   lm(data = ., C6H6.GT. ~ PT08.S1.CO.)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S1.CO., data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0888 -1.6245  0.0254  1.6468  9.3398
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.374e+01  4.790e-01  -49.56   <2e-16 ***
## PT08.S1.CO.  2.857e-02  3.888e-04   73.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.702 on 825 degrees of freedom
## Multiple R-squared:  0.8674, Adjusted R-squared:  0.8673
## F-statistic:  5399 on 1 and 825 DF,  p-value: < 2.2e-16
```

```
air%>%
   ggplot(aes(x = PT08.S1.CO., y = C6H6.GT.))+
```

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

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



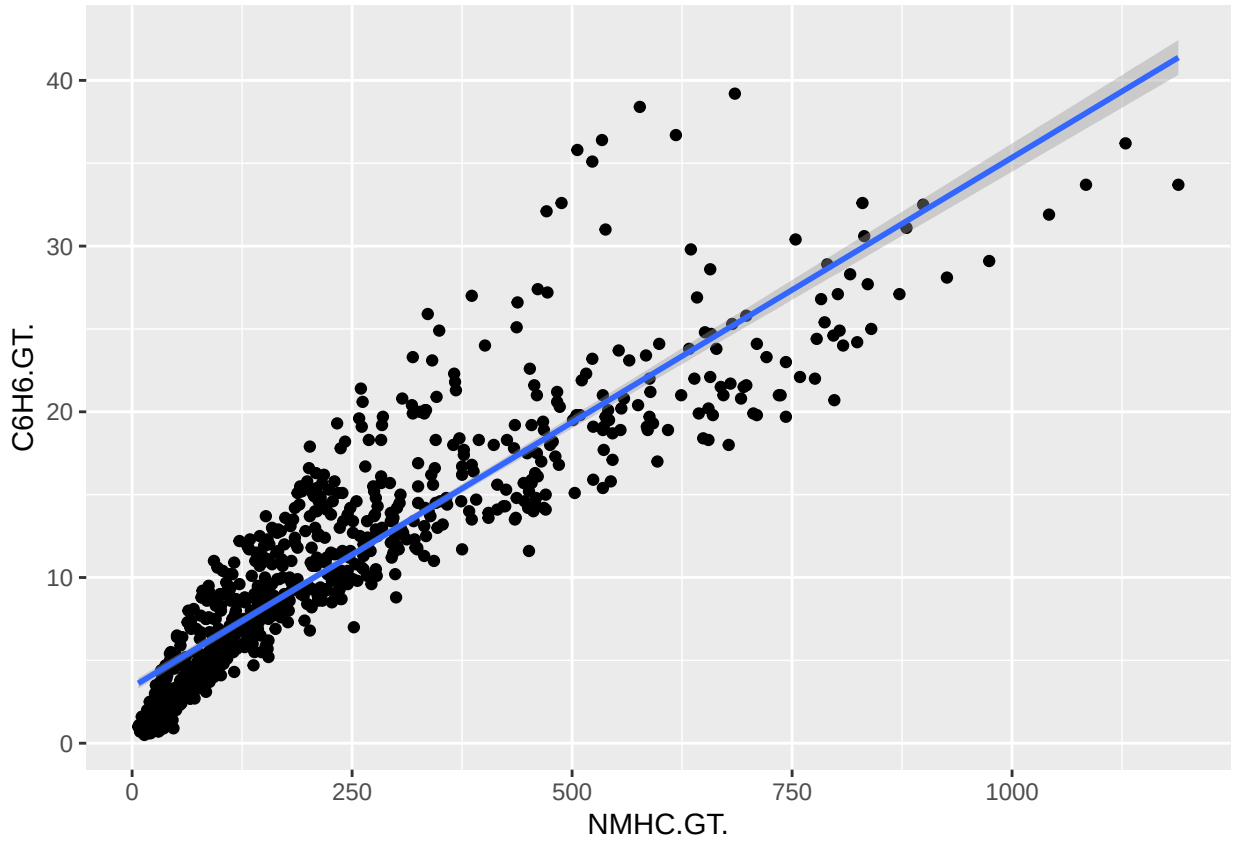

```
air%>%
   lm(data = ., C6H6.GT. ~ NMHC.GT.)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ NMHC.GT., data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1876 -2.0558 -0.6626  1.3815 16.5740
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.3891818  0.1696364   19.98   <2e-16 ***
## NMHC.GT.    0.0319528  0.0005453   58.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 3.267 on 825 degrees of freedom
## Multiple R-squared:  0.8063, Adjusted R-squared:  0.806
## F-statistic:  3434 on 1 and 825 DF,  p-value: < 2.2e-16
```

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

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



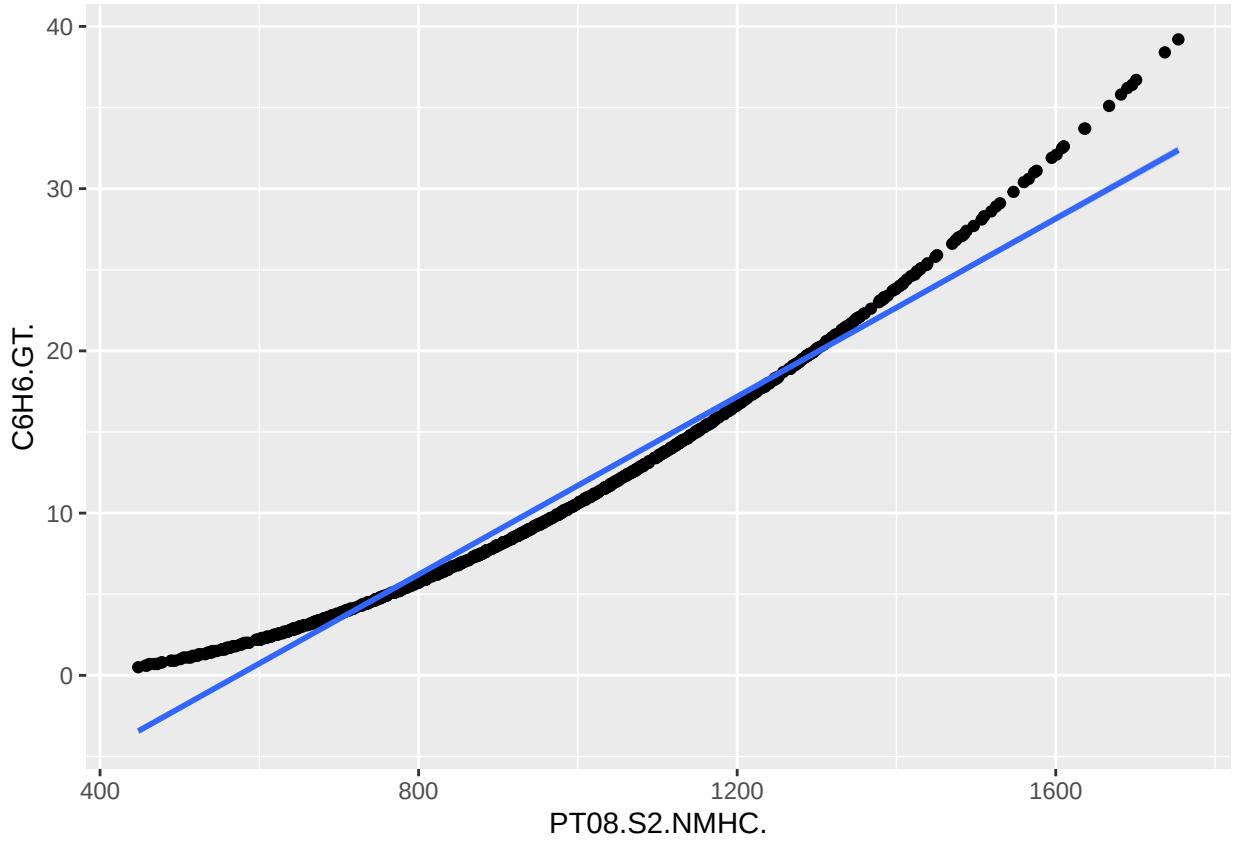

```
air%>%
   lm(data = ., C6H6.GT. ~ PT08.S2.NMHC.)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S2.NMHC., data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1470 -0.9581 -0.4612  0.5492  6.8243
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.572e+01  1.685e-01  -93.27   <2e-16 ***
## PT08.S2.NMHC.  2.742e-02  1.682e-04  163.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.288 on 825 degrees of freedom
## Multiple R-squared:  0.9699, Adjusted R-squared:  0.9699
## F-statistic: 2.658e+04 on 1 and 825 DF,  p-value: < 2.2e-16
```

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

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



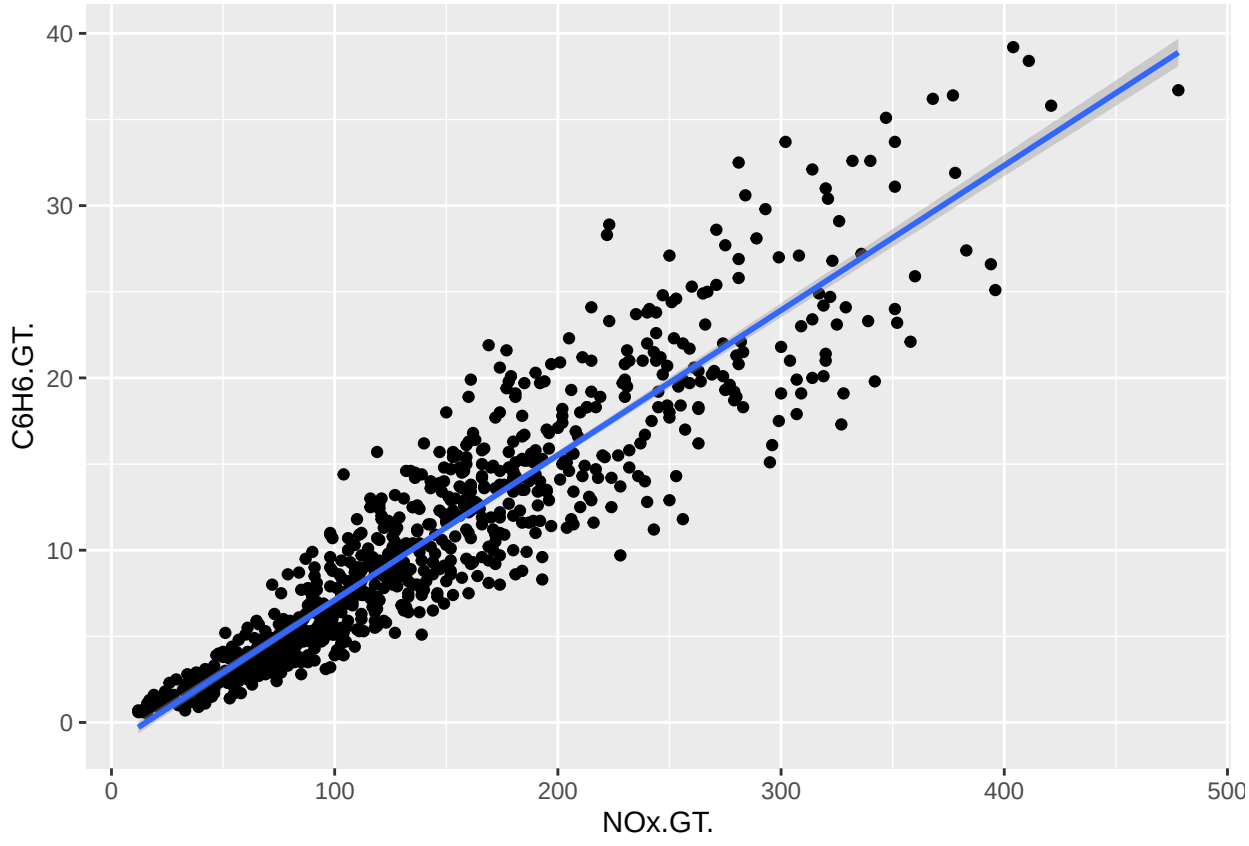

```
air%>%
   lm(data = ., C6H6.GT. ~ NOx.GT.)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ NOx.GT., data = .)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8965 -1.5222 -0.1907  1.2497 11.4460
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.292115   0.195126  -6.622 6.39e-11 ***
## NOx.GT.      0.084063   0.001181  71.157  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.778 on 825 degrees of freedom
## Multiple R-squared:  0.8599, Adjusted R-squared:  0.8597
## F-statistic:  5063 on 1 and 825 DF,  p-value: < 2.2e-16
```

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

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

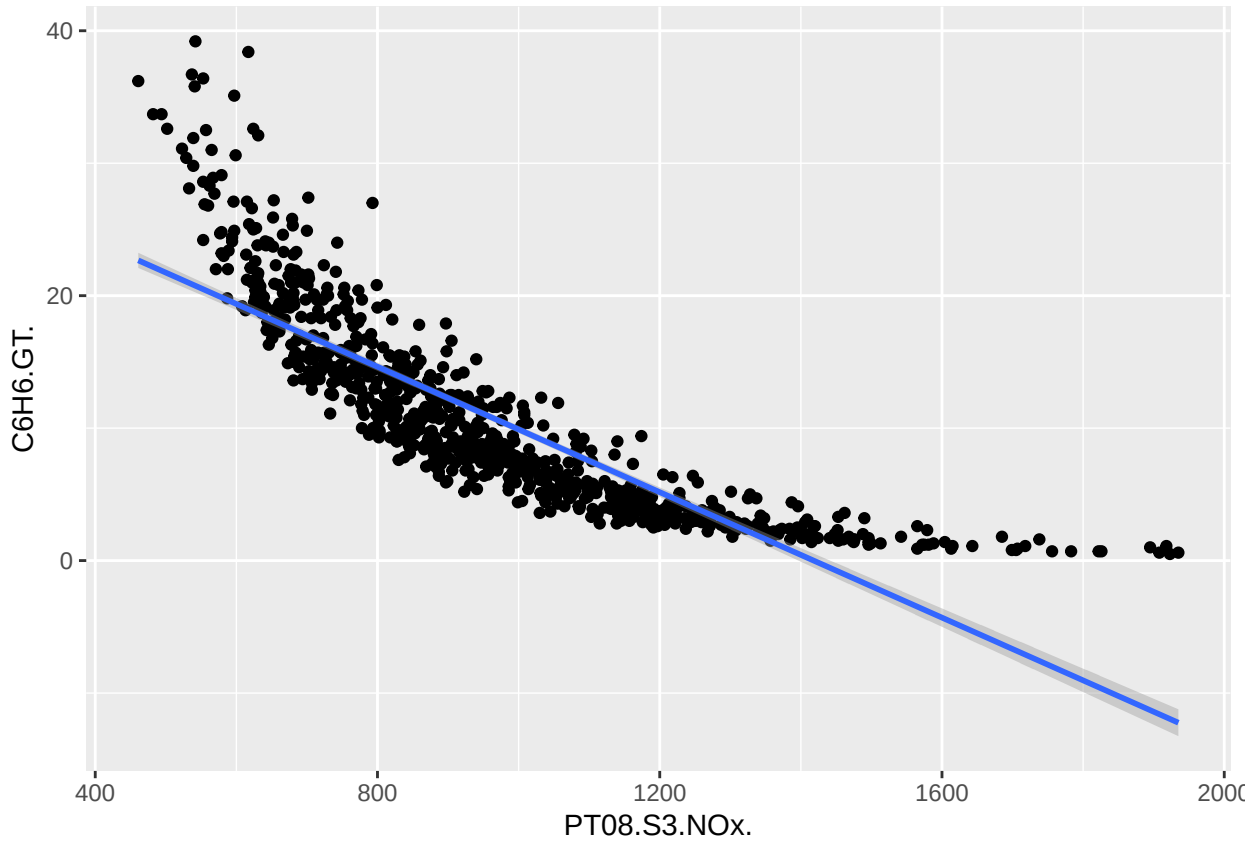


```r
air%>%
   lm(data = ., C6H6.GT. ~ PT08.S3.NOx.)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S3.NOx., data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5253 -2.6883 -0.9271  1.7269 19.4285
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.5821174  0.5130616   65.45   <2e-16 ***
## PT08.S3.NOx. -0.0236801  0.0005134  -46.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.924 on 825 degrees of freedom
## Multiple R-squared:  0.7205, Adjusted R-squared:  0.7202
## F-statistic:  2127 on 1 and 825 DF,  p-value: < 2.2e-16
```

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

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



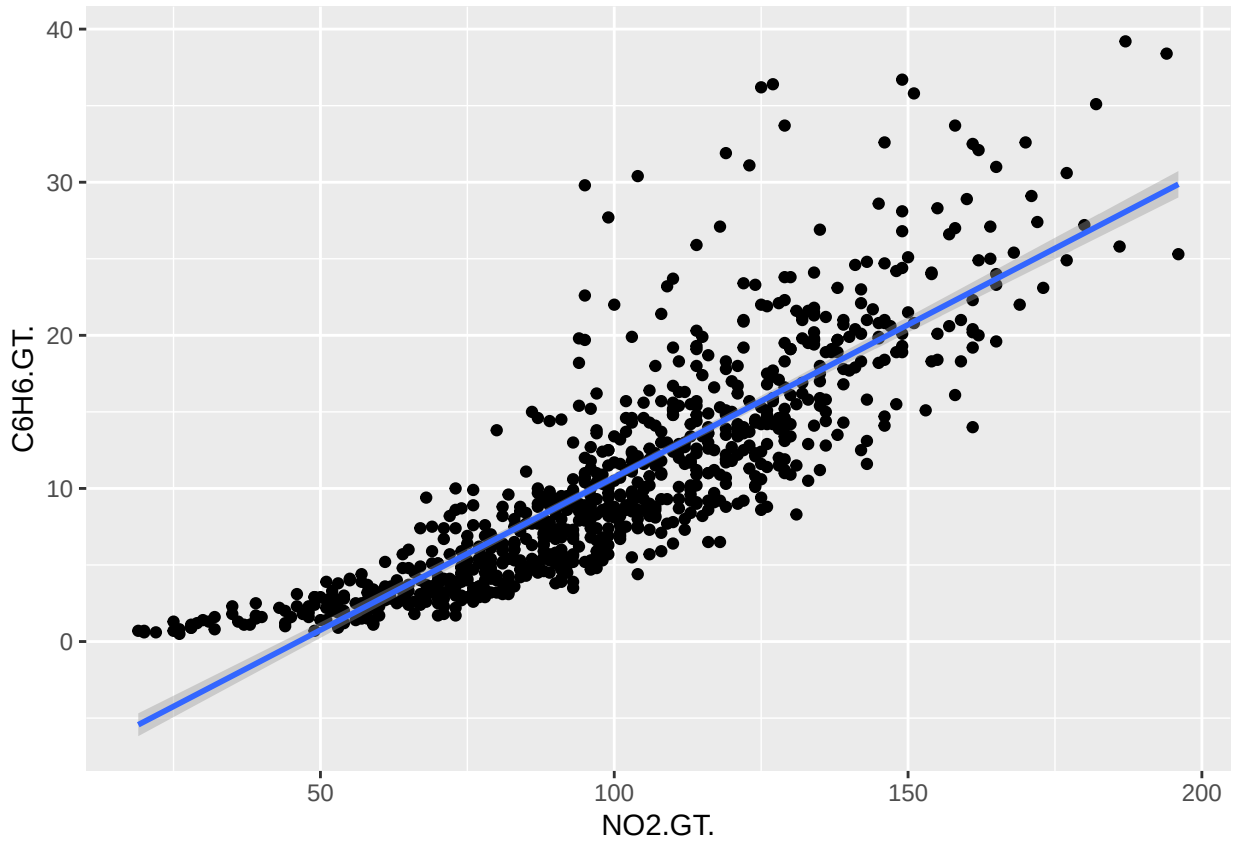

```
air%>%
   lm(data = ., C6H6.GT. ~ NO2.GT.)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ NO2.GT., data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8853 -2.5243 -0.5853  1.7552 20.4947
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.225139   0.458452  -20.12   <2e-16 ***
## NO2.GT.      0.199444   0.004363   45.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.949 on 825 degrees of freedom
## Multiple R-squared:  0.717,  Adjusted R-squared:  0.7166
## F-statistic:  2090 on 1 and 825 DF,  p-value: < 2.2e-16
```

```
air%>%
   ggplot(aes(x = NO2.GT., y = C6H6.GT.))+
```

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

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



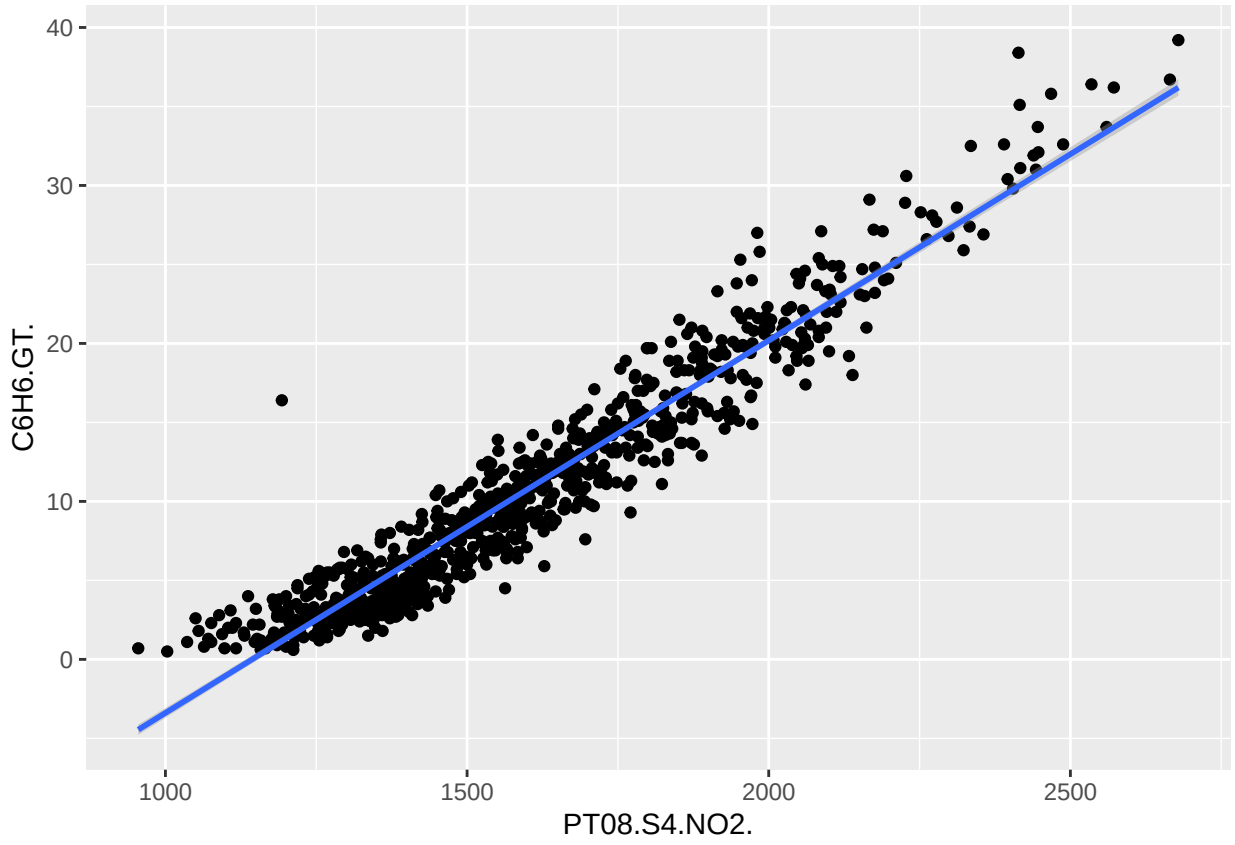

```
air%>%
   lm(data = ., C6H6.GT. ~ PT08.S4.NO2.)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S4.NO2., data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5167 -1.4177  0.0103  1.1915 15.2398
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.697e+01  3.858e-01  -69.91   <2e-16 ***
## PT08.S4.NO2.  2.358e-02  2.368e-04   99.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.058 on 825 degrees of freedom
## Multiple R-squared:  0.9232, Adjusted R-squared:  0.9231
## F-statistic:  9911 on 1 and 825 DF,  p-value: < 2.2e-16
```

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

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



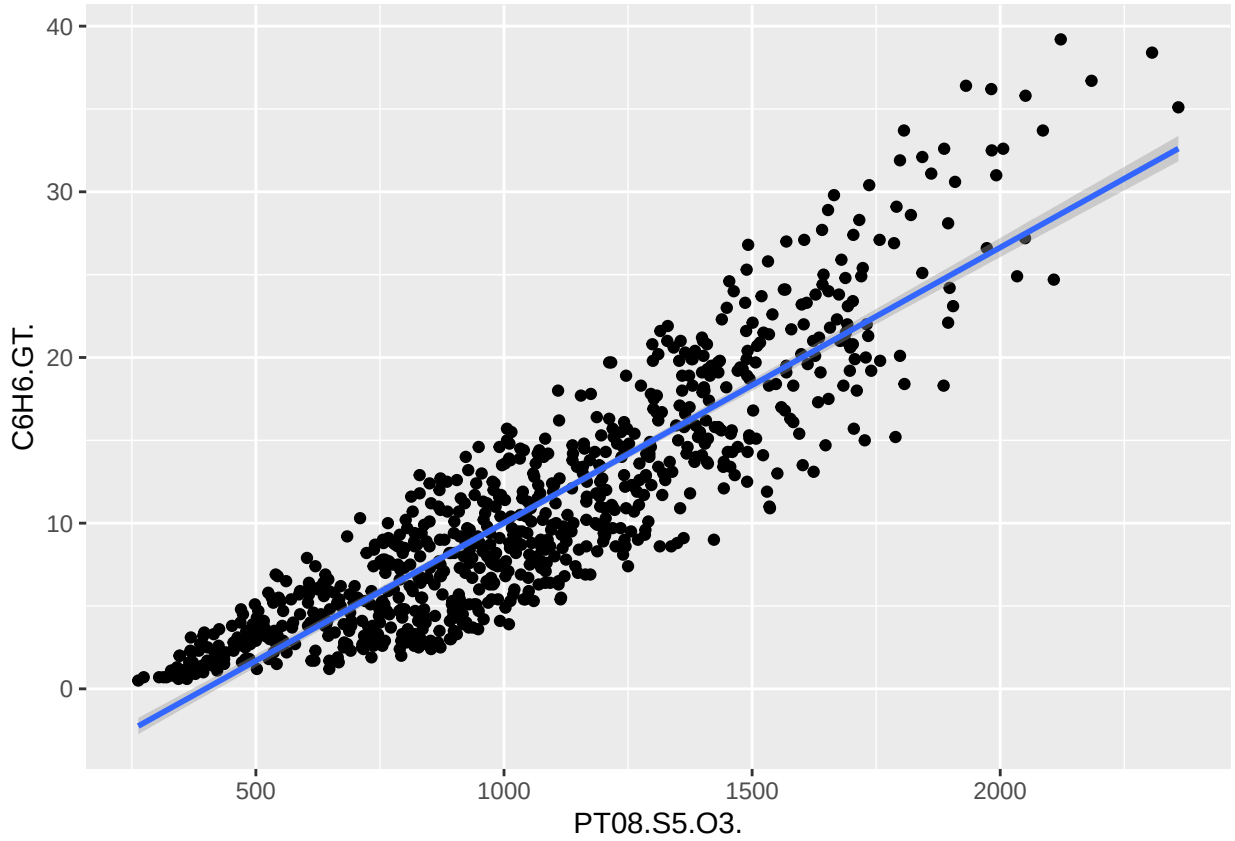

```
air%>%
   lm(data = .,C6H6.GT.  ~ PT08.S5.O3.)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S5.O3., data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0434 -2.5352  0.2444  2.1773 10.9090
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.6198822  0.3194802  -20.72   <2e-16 ***
## PT08.S5.O3.  0.0166292  0.0002853   58.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.281 on 825 degrees of freedom
## Multiple R-squared:  0.8046, Adjusted R-squared:  0.8043
## F-statistic:  3396 on 1 and 825 DF,  p-value: < 2.2e-16
```

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

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



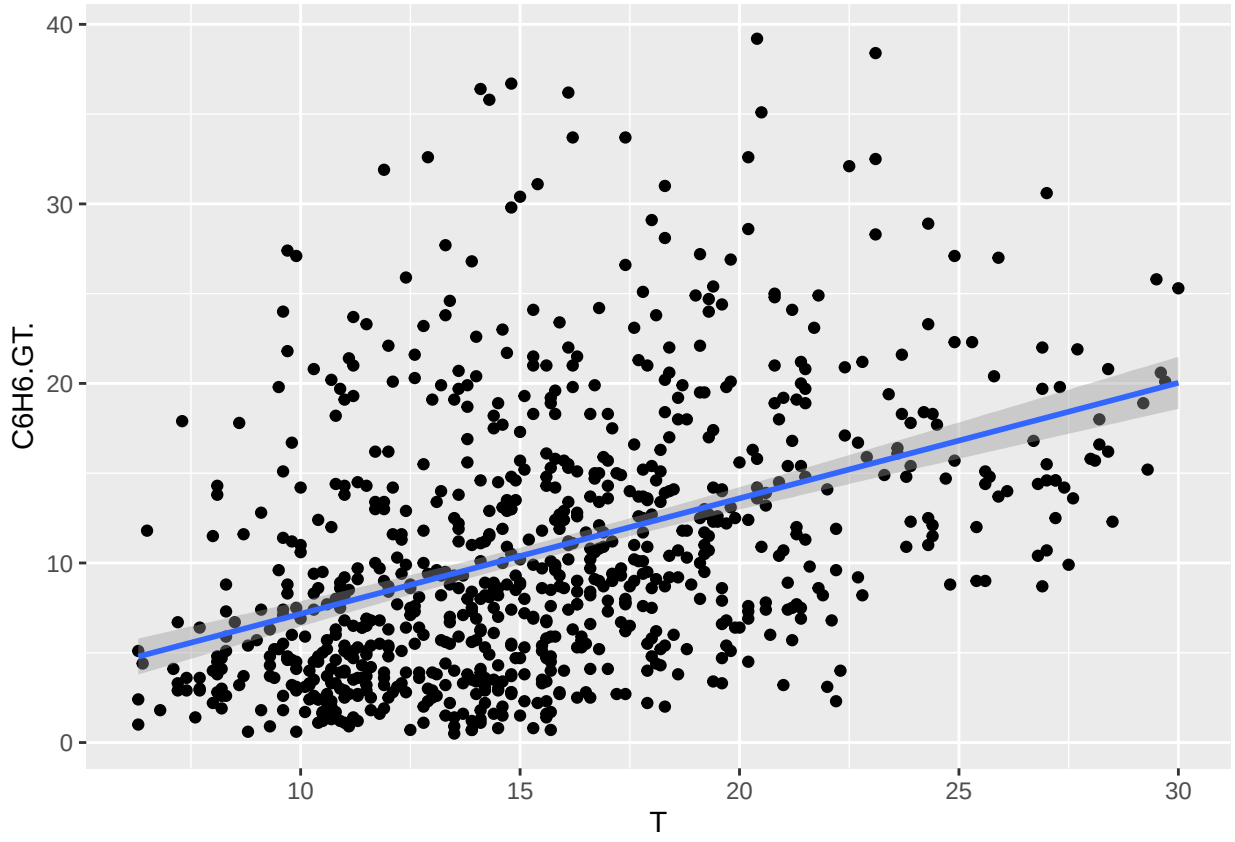

```
air%>%
   lm(data = .,C6H6.GT. ~ `T`)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ T, data = .)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.716  -4.813  -1.526   3.075  26.595
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.73568    0.79384   0.927    0.354
## T            0.64324    0.04861  13.232   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.742 on 825 degrees of freedom
## Multiple R-squared:  0.1751, Adjusted R-squared:  0.1741
## F-statistic: 175.1 on 1 and 825 DF,  p-value: < 2.2e-16
```

```
air%>%
   ggplot(aes(x = `T`, y = C6H6.GT.))+
  geom_point()+
  geom_smooth(method = lm)
```

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

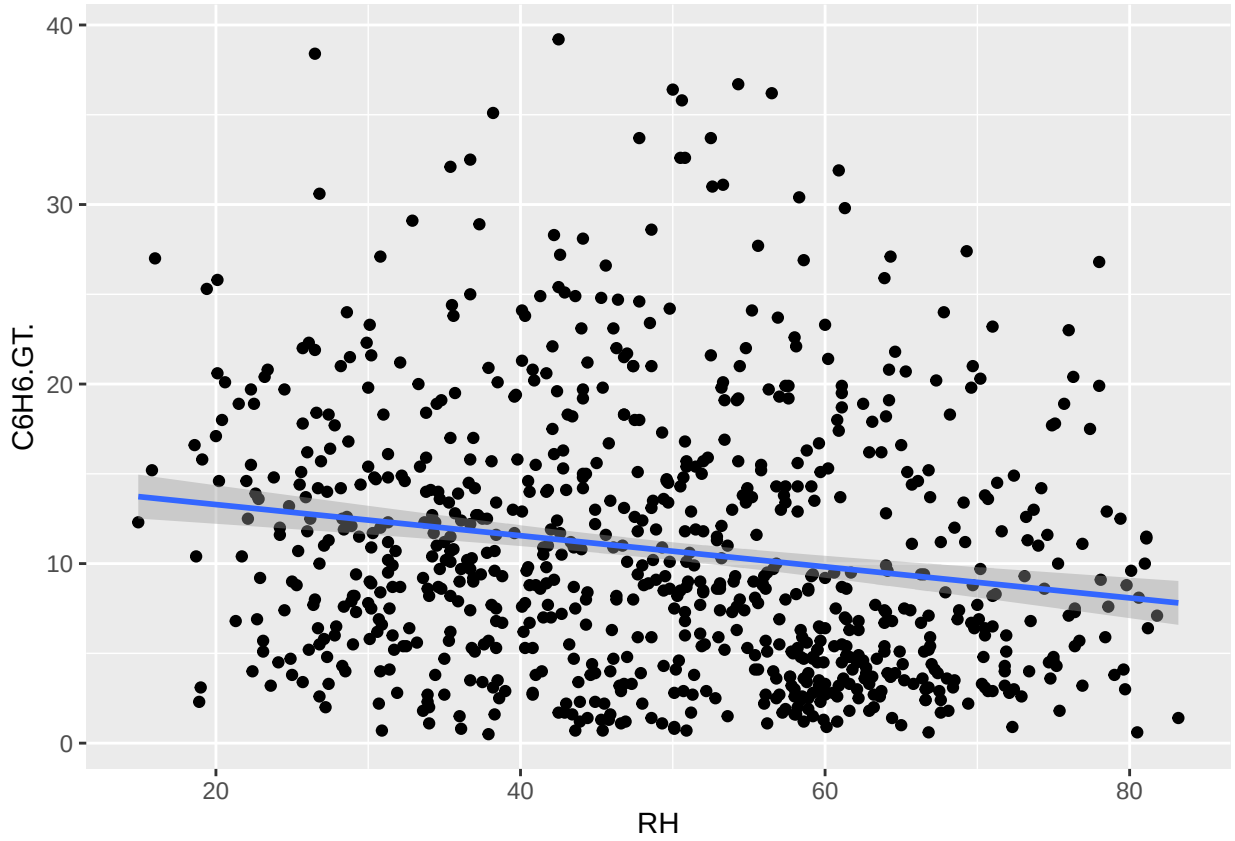


```r
air%>%
   lm(data = ., C6H6.GT. ~ RH)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ RH, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.645  -5.566  -1.584   3.962  27.861
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.02325    0.85505  17.570  < 2e-16 ***
## RH          -0.08669    0.01665  -5.208 2.41e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.304 on 825 degrees of freedom
## Multiple R-squared:  0.03183,    Adjusted R-squared:  0.03066
## F-statistic: 27.12 on 1 and 825 DF,  p-value: 2.412e-07
```

```r
air%>%
   ggplot(aes(x = RH, y = C6H6.GT.))+
  geom_point()+
  geom_smooth(method = lm)
```

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



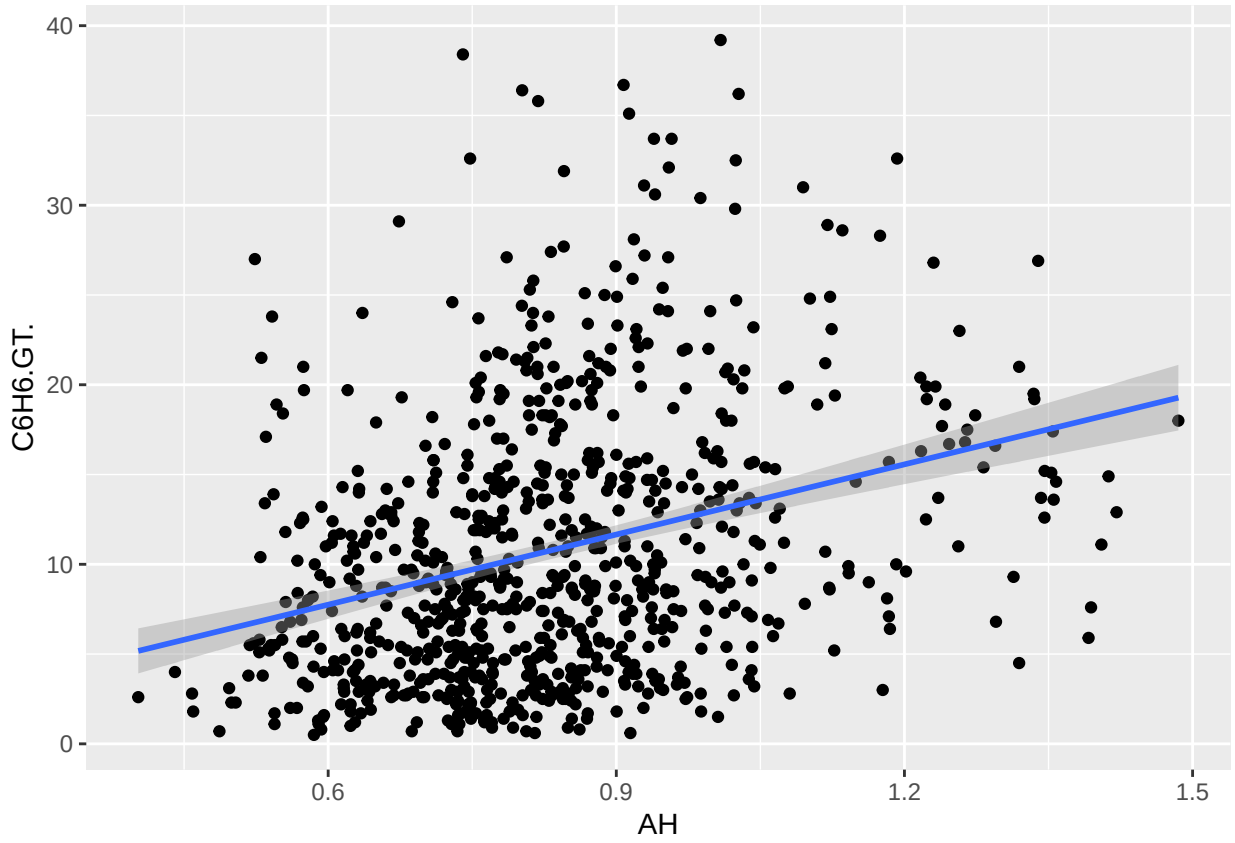

```
air%>%
   lm(data = ., C6H6.GT. ~ AH)%>%
  summary()
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ AH, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.621  -5.372  -1.530   3.850  28.821
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.06339    1.16889  -0.054    0.957
## AH          13.02454    1.37393   9.480   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.049 on 825 degrees of freedom
## Multiple R-squared:  0.09823,    Adjusted R-squared:  0.09714
## F-statistic: 89.87 on 1 and 825 DF,  p-value: < 2.2e-16
```

```
air%>%
   ggplot(aes(x = AH, y = C6H6.GT.))+
```

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

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



I've choosed benzoate dependence on CO and on tin oxide. Separated data in 75:25% We can see that R^2 is close to 1 which says about linear dependence. P-value is close to 0 - independent variables explain the dynamics of the dependent variable.

```
set.seed(2)
sep <- sample.int(n = nrow(air), size = floor(.75*nrow(air)))
train <- air[sep,]
test <- air[-sep,]

train1 <- train[,c('C6H6.GT.', 'CO.GT.')]

test1 <- test[,c('C6H6.GT.', 'CO.GT.')]

mod1 <- lm(data = train1,C6H6.GT. ~ CO.GT.)

a1 <- summary(mod1)
a1
```

```
##
## Call:
```

```
## lm(formula = C6H6.GT. ~ CO.GT., data = train1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5199 -0.9377 -0.0730  0.7945  6.0076
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.27550    0.13376  -9.536   <2e-16 ***
## CO.GT.       5.11590    0.04896 104.499   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.731 on 618 degrees of freedom
## Multiple R-squared:  0.9464, Adjusted R-squared:  0.9464
## F-statistic: 1.092e+04 on 1 and 618 DF,  p-value: < 2.2e-16
```

```
a1$stat <- cor.test(train1$CO.GT., train1$C6H6.GT.)
pred1 <- predict(mod1, newdata = test1)
head(pred1)
```

```
##         7        13        17        18        20        21
##  4.863570  2.305622  7.421518  6.398339  8.444697 13.560593
```

```
test1$C6H6.GT._pred <- pred1
head(test1)
```

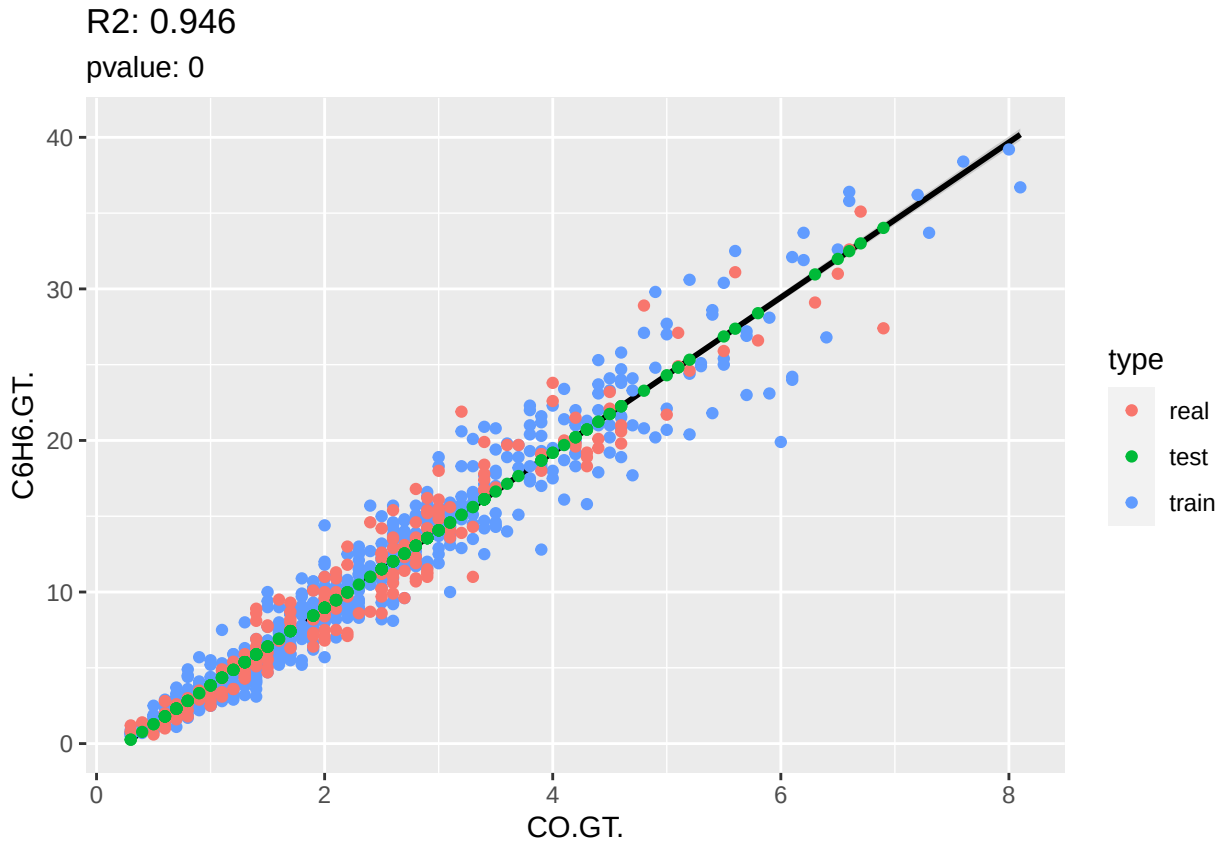
```
##    C6H6.GT. CO.GT. C6H6.GT._pred
## 7       3.6    1.2      4.863570
## 13      1.6    0.7      2.305622
## 17      6.3    1.7      7.421518
## 18      5.0    1.5      6.398339
## 20      7.3    1.9      8.444697
## 21     11.5    2.9     13.560593
```

```
test1 <- rbindlist(list(test1[,c(2, 1)], test1[,c(2,3)]))
```

```
## Column 2 ['C6H6.GT._pred'] of item 2 is missing in item 1. Use fill=TRUE to fill with NA (NULL for li
```

```
train1$type <- 'train'
test1$type <- 'test'
test1[1:(nrow(test1)/2),3] <- 'real'
all <- rbind(train1, test1)

ggplot(data = all, aes(x = CO.GT., y = C6H6.GT., color = type))+
  geom_smooth(method = 'lm', color = 'black')+
  ggtitle(paste("R2", round(a1$r.squared, 3),
                                  sep = ": "),
                           paste("pvalue", a1$stat$p.value, sep = ": "))+
  geom_point()
```

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




```
train2 <- train[,c('PT08.S1.CO.', 'C6H6.GT.')]
test2 <- test[,c('PT08.S1.CO.', 'C6H6.GT.')]
mod2 <- lm(data = train2, C6H6.GT.~PT08.S1.CO.)
a2 <- summary(mod2)
a2
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S1.CO., data = train2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8329 -1.6542 -0.0321  1.6509  8.9618
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.428e+01  5.397e-01  -45.00   <2e-16 ***
## PT08.S1.CO.  2.909e-02  4.405e-04   66.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.636 on 618 degrees of freedom
## Multiple R-squared:  0.8759, Adjusted R-squared:  0.8757
## F-statistic:  4362 on 1 and 618 DF,  p-value: < 2.2e-16
```

```
a2$stat <- cor.test(train2$PT08.S1.CO., train2$C6H6.GT.)

pred2 <- predict(mod2, newdata = test2)
head(pred2)
```

```
##         7        13        17        18        20        21
## 10.189920  6.320608 11.586363 10.015365 13.128270 15.601138
```

```
test2$C6H6.GT._pred <- pred2
head(test2)
```

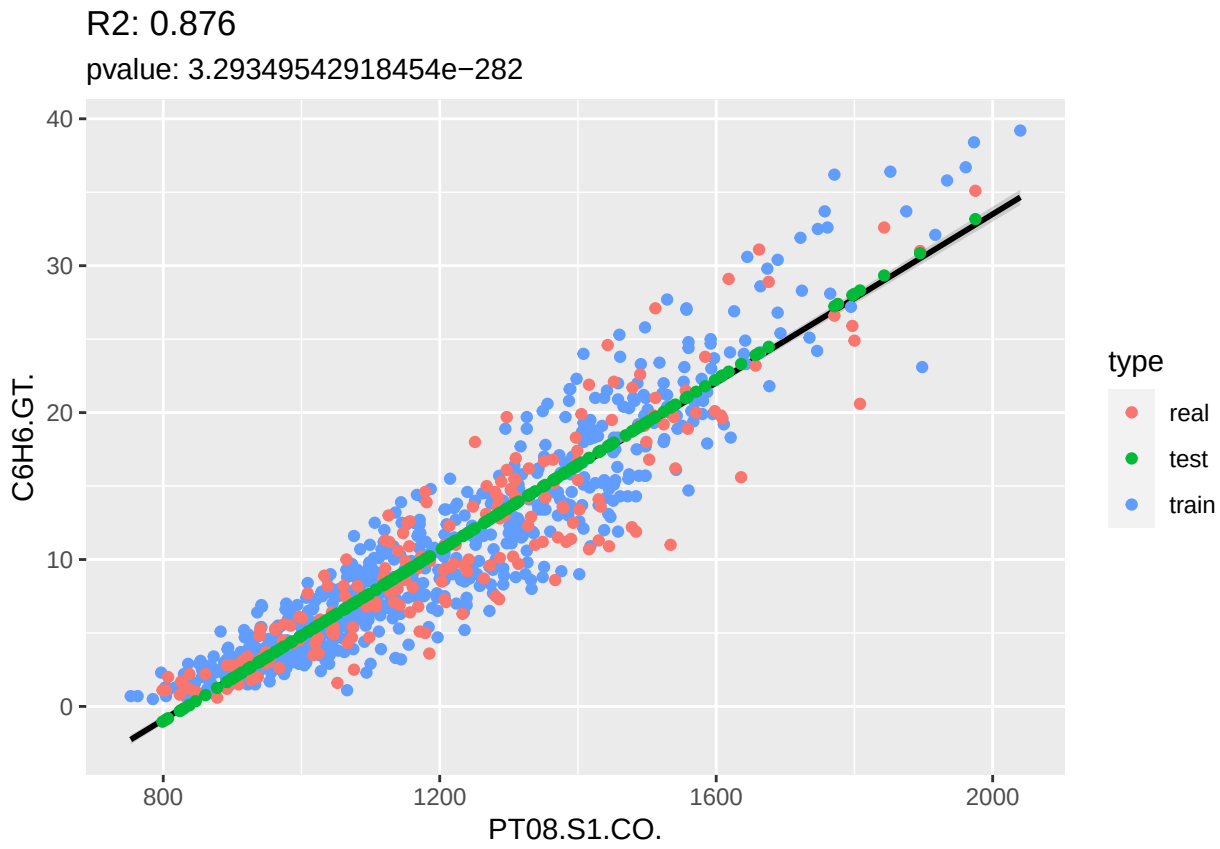
```
##    PT08.S1.CO. C6H6.GT. C6H6.GT._pred
## 7         1185      3.6     10.189920
## 13        1052      1.6      6.320608
## 17        1233      6.3     11.586363
## 18        1179      5.0     10.015365
## 20        1286      7.3     13.128270
## 21        1371     11.5     15.601138
```

```
test2 <- rbindlist(list(test2[,c(1,2)], test2[,c(1,3)]))
```

```
## Column 2 ['C6H6.GT._pred'] of item 2 is missing in item 1. Use fill=TRUE to fill with NA (NULL for li
```

```
train2$type <- 'train'
test2$type <- 'test'
test2[1:(nrow(test2)/2),3] <- 'real'
all2 <- rbind(train2[,c(2,1,3)], test2)

ggplot(data = all2, aes(x = PT08.S1.CO., y = C6H6.GT., color = type))+
  geom_smooth(method = 'lm', color = 'black')+
  ggtitle(paste("R2", round(a2$r.squared, 3),
                                  sep = ": "),
                          paste("pvalue", a2$stat$p.value, sep = ": "))+
   geom_point()
```

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


R2: 0.876
pvalue: 3.29349542918454e−282
C6H6.GT.
PT08.S1.CO.
type
real
test
train

‘‘