

Statistics in R: Task 1

Liuaza Etezova

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

Anscombe

```
data(anscombe)
head(anscombe)
```

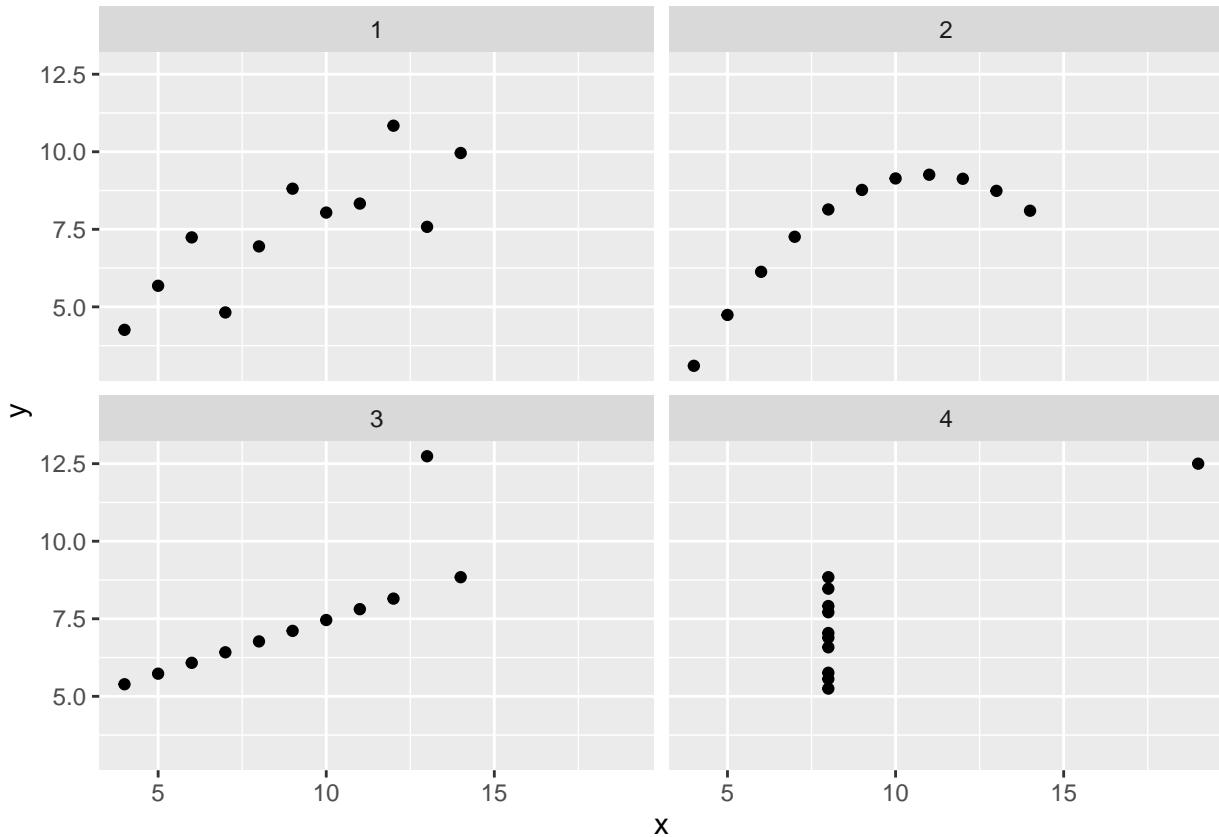
```
##   x1 x2 x3 x4   y1   y2   y3   y4
## 1 10 10 10 10  8.04 9.14 7.46 6.58
## 2  8   8   8   8  6.95 8.14 6.77 5.76
## 3 13 13 13 13  8 7.58 8.74 12.74 7.71
## 4  9   9   9   9  8 8.81 8.77 7.11 8.84
## 5 11 11 11 11  8 8.33 9.26 7.81 8.47
## 6 14 14 14 14  8 9.96 8.10 8.84 7.04
```

```
anscombeL <- data.frame(
  set = rep(1:4, each=11),
  x = unlist(anscombe[, c(1:4)]),
  y = unlist(anscombe[, c(5:8)]))
rownames(anscombeL) <- NULL
head(anscombeL)
```

```
##   set   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
```

Scatter plot faceted by set

```
ggplot(data=anscombeL, mapping=aes(x, y)) +
  geom_point() +
  facet_wrap(~set)
```



Summary calculation (mean, sd) grouped by set

```
by_set <- anscombeL %>%
  group_by(set)

by_set %>%
  summarise_all(list(~mean(.), ~sd(.)))

## # A tibble: 4 x 5
##   set   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
```

Correlations by set

```
by_set %>%
  summarise(cor_pearson = cor(x, y, method="pearson"),
            cor_kendall = cor(x, y, method="kendall"),
            cor_spearman = cor(x, y, method="spearman"))

## # A tibble: 4 x 4
##       set cor_pearson cor_kendall cor_spearman
##   <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

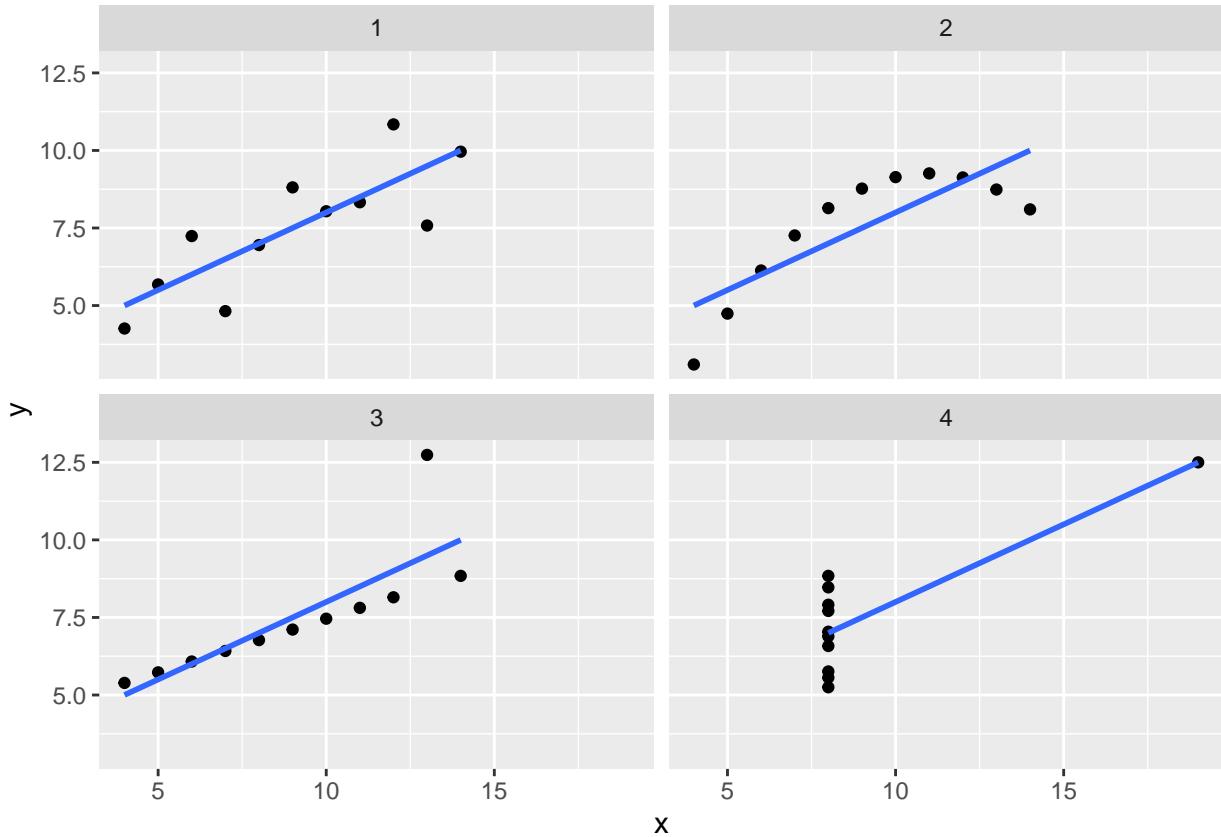
by_set %>%
  summarise(p_pearson = cor.test(x, y, method="pearson")$p.value,
            p_kendall = cor.test(x, y, method="kendall")$p.value,
            p_spearman = cor.test(x, y, method="spearman")$p.value)

## # A tibble: 4 x 4
##       set p_pearson p_kendall p_spearman
##   <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
```

Scatter plot faceted by set with geom_smooth()

```
ggplot(data=anscombeL, mapping=aes(x, y)) +
  geom_point() +
  geom_smooth(method='lm', se=F) +
  facet_wrap(~set)

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



Air quality

```
air_quality <- read.csv('AirQualityUCI.csv', sep=';')
str(air_quality)
```

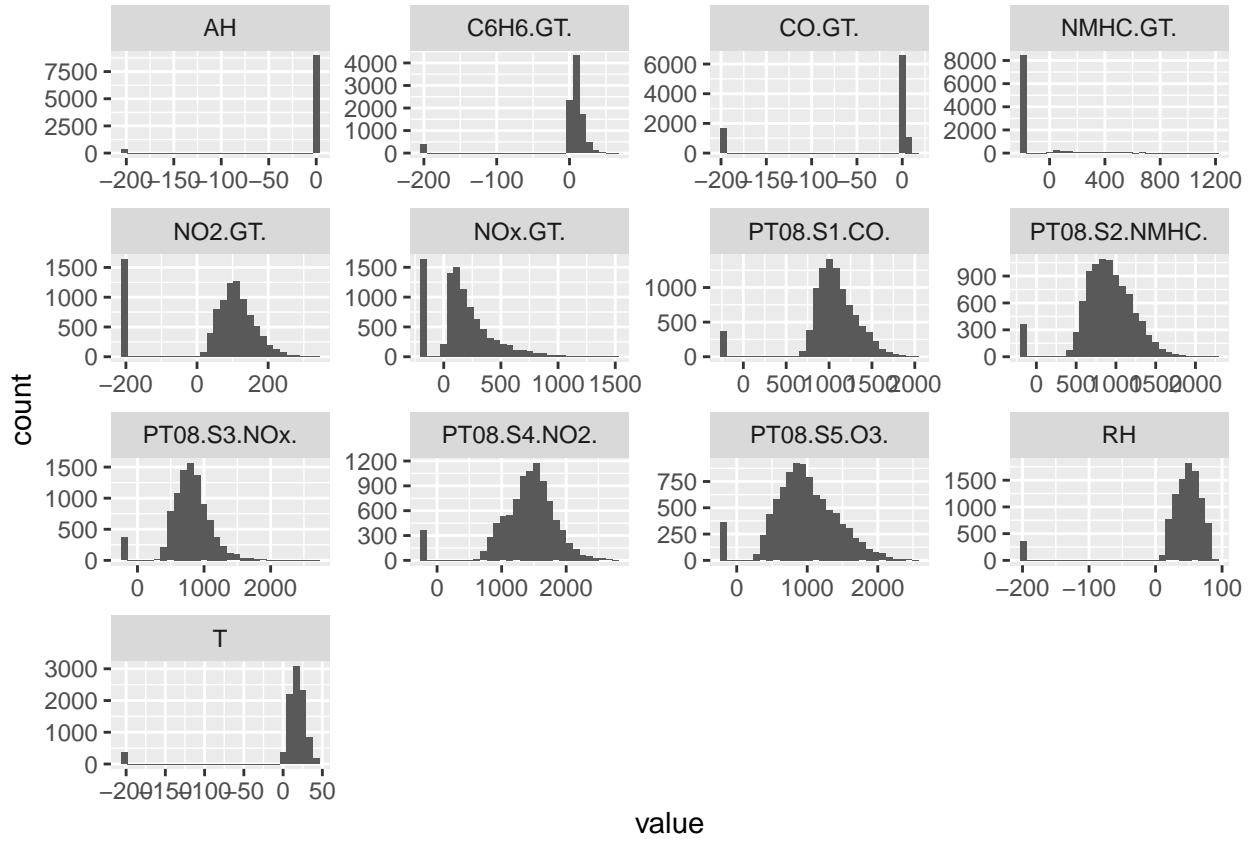
```
## 'data.frame': 9471 obs. of 17 variables:
## $ Date      : Factor w/ 392 levels "", "01/01/2005", ...
## $ Time      : Factor w/ 25 levels "", "00.00.00", ...
## $ CO.GT.    : Factor w/ 105 levels "", "-200", "-200,0", ...
## $ PT08.S1.CO.: int 1360 1292 1402 1376 1272 1197 1185 1136 1094 1010 ...
## $ NMHC.GT.  : int 150 112 88 80 51 38 31 31 24 19 ...
## $ C6H6.GT.  : Factor w/ 409 levels "", "-200,0", "0,1", ...
## $ 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.S5.03.: int 1268 972 1074 1203 1110 949 733 730 620 501 ...
## $ T         : Factor w/ 438 levels "", "-0,1", "-0,2", ...
## $ RH        : Factor w/ 755 levels "", "-200", "10,0", ...
## $ AH        : Factor w/ 6685 levels "", "-200", "0,1847", ...
## $ X         : logi NA NA NA NA NA NA ...
## $ X.1       : logi NA NA NA NA NA NA ...
```

Tidying

```
air_quality <- air_quality[-c(9358:nrow(air_quality)),  
                           !(names(air_quality) %in% c('X', 'X.1'))]  
  
air_quality$Date <- as.Date(as.character(air_quality$Date), "%d/%m/%Y")  
air_quality$Time <- chron(times=air_quality$Time, format=c('h.m.s'))  
  
columns <- c('CO.GT.', 'C6H6.GT.', 'T', 'RH', 'AH')  
air_quality[columns] <- lapply(air_quality[columns], function(x) gsub(", ", ".", x))  
air_quality[columns] <- lapply(air_quality[columns], as.numeric)  
  
str(air_quality)  
  
## 'data.frame': 9357 obs. of 15 variables:  
## $ Date : Date, format: "2004-03-10" "2004-03-10" ...  
## $ Time : 'times' num 18.00.00 19.00.00 20.00.00 21.00.00 22.00.00 ...  
## ..- attr(*, "format")= chr "h.m.s"  
## $ 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.S5.03. : 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 ...
```

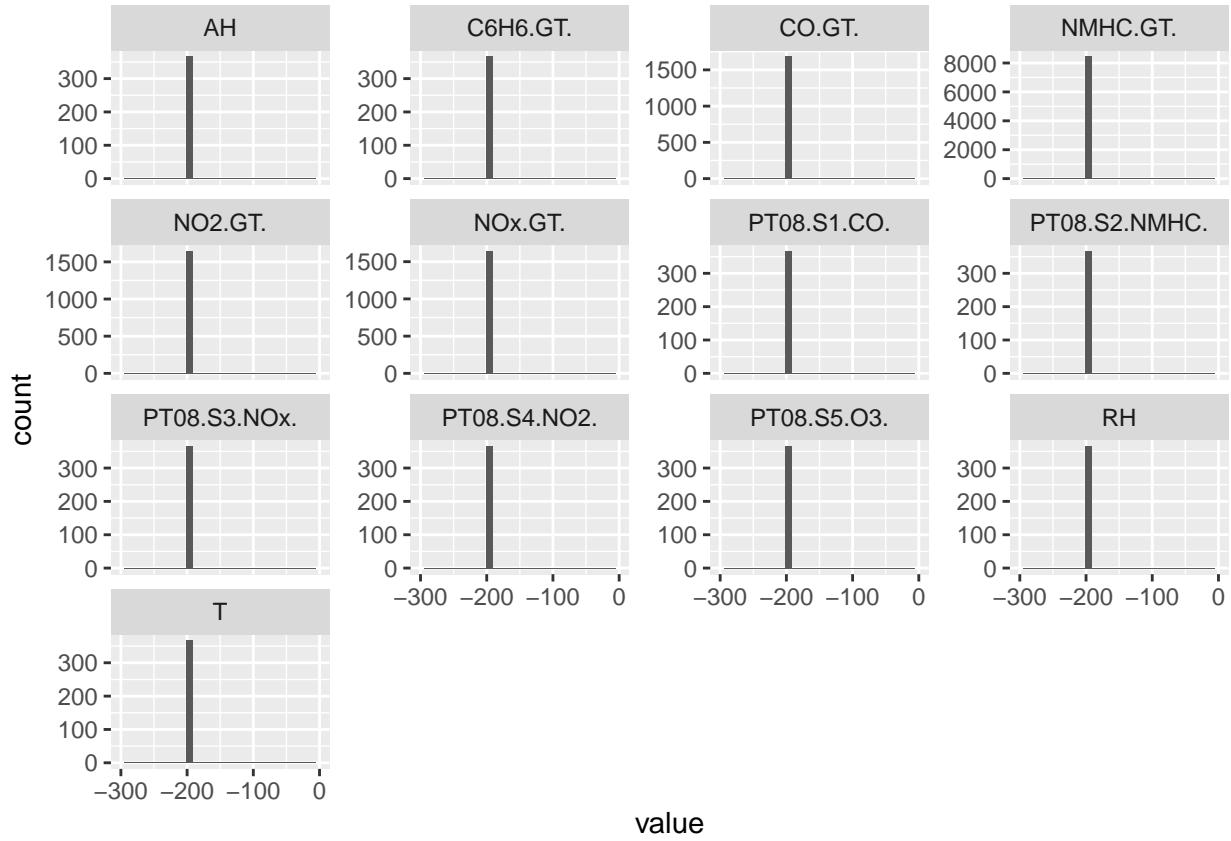
Checking each variable

```
airq_for_hist <- pivot_longer(air_quality, cols=CO.GT.:AH)  
ggplot(airq_for_hist, aes(value)) +  
  geom_histogram() +  
  facet_wrap(~name, scales='free')  
  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(airq_for_hist, aes(value)) +
  geom_histogram() +
  facet_wrap(~name, scales='free_y') +
  xlim(-300, 0)
```

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



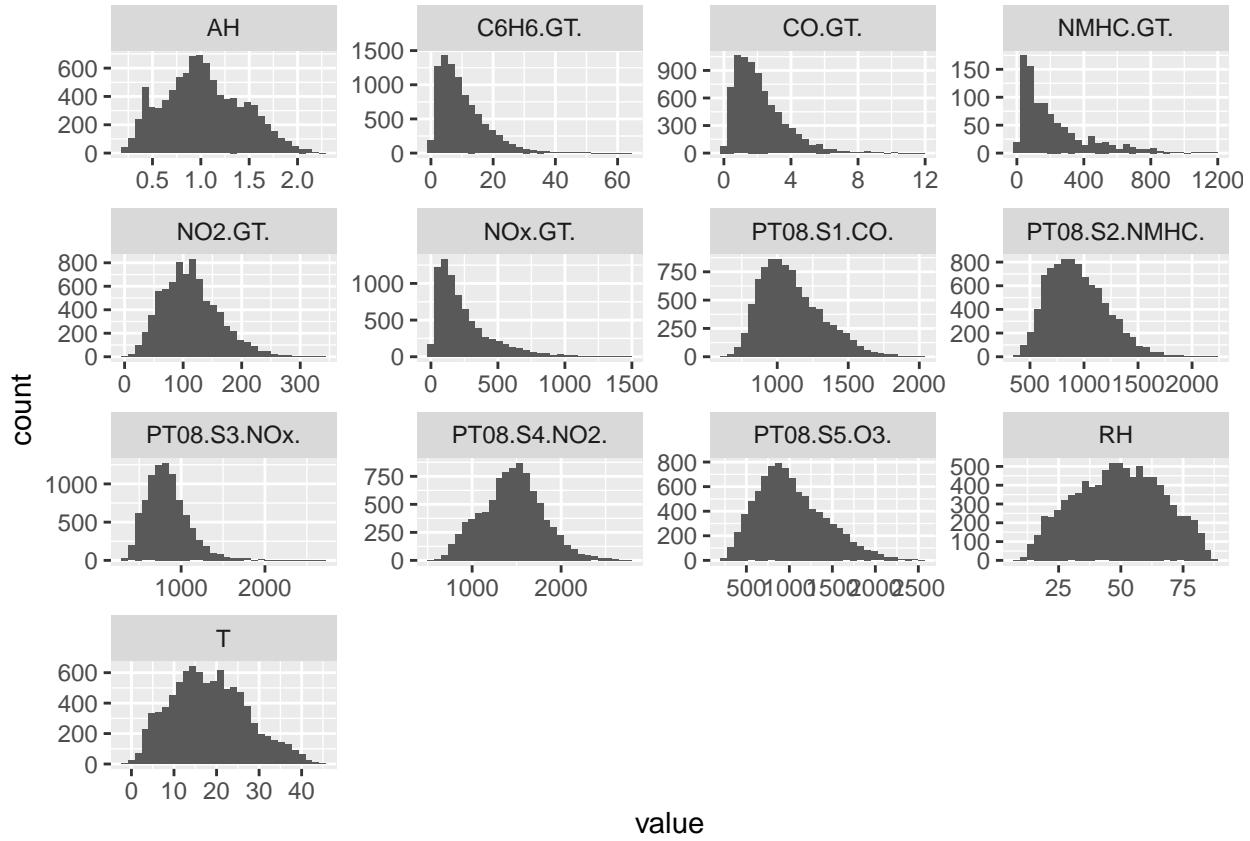
-200 like the default value for this data. I will not delete all rows with this value because this way I will lose too much data (8,530 rows out of 9,357), but I will replace all individual values with NAs to pairwise exclude them in correlation analysis (“use=‘pairwise.complete.obs’” argument in “corr” function).

Cleaning

```
air_quality[air_quality == -200] <- NA

airq_for_hist <- pivot_longer(air_quality, cols=CO.GT.:AH)
ggplot(airq_for_hist, aes(value)) +
  geom_histogram() +
  facet_wrap(~name, scales='free')

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



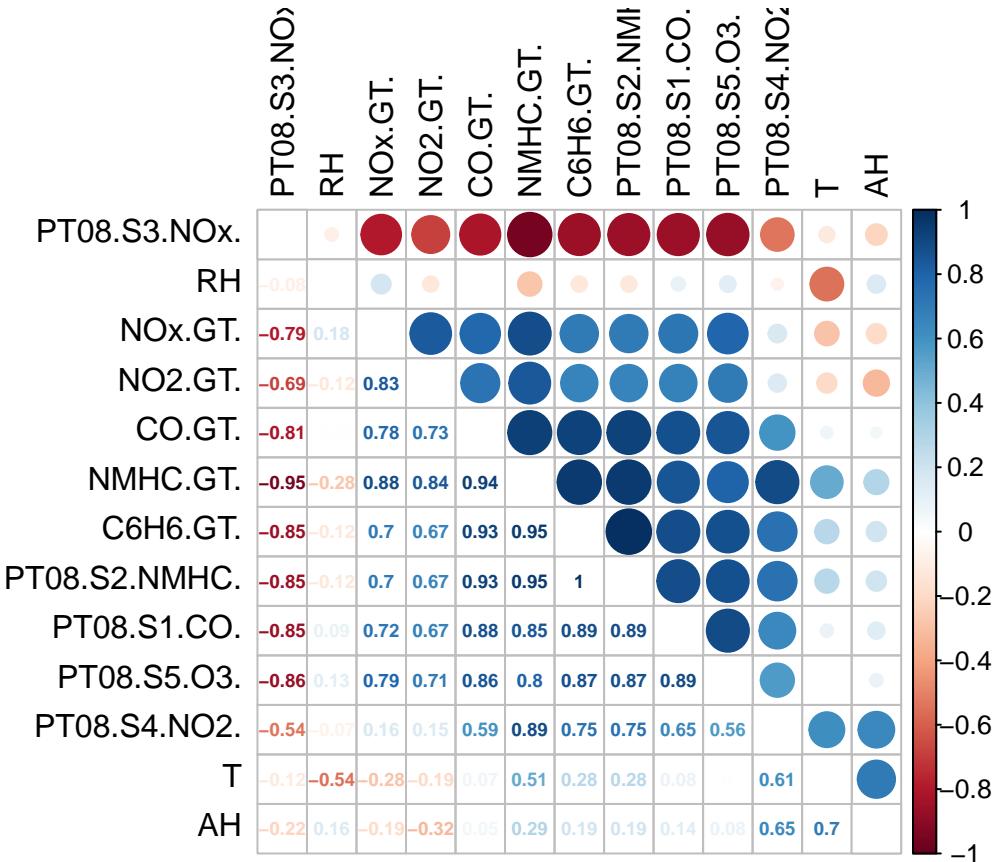
Cross correlations

```

cors <- cor(air_quality[3:15], use='pairwise.complete.obs', method='spearman')

corrplot.mixed(cors, order='hclust', lower='number', tl.pos='lt', tl.col='black',
               number.cex=0.55)

```

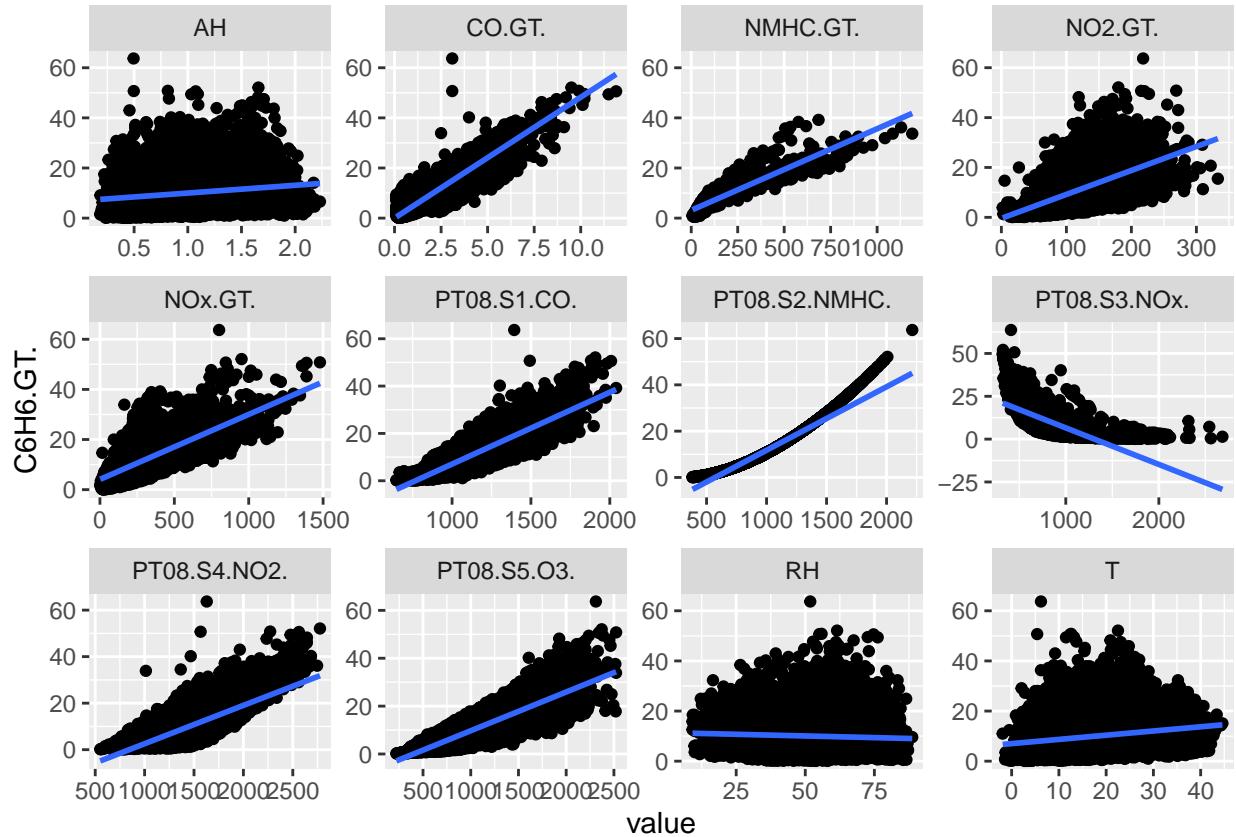


Linear models, response - C6H6(GT)

Checking the linearity of dependencies

```
airq_for_lm <- pivot_longer(air_quality, cols=-c('Date','Time', 'C6H6.GT.'))
ggplot(airq_for_lm, aes(x=value, y=C6H6.GT.)) +
  geom_point() +
  geom_smooth(method='lm', se=F) +
  facet_wrap(~name, scales='free')
```

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



Linearizing

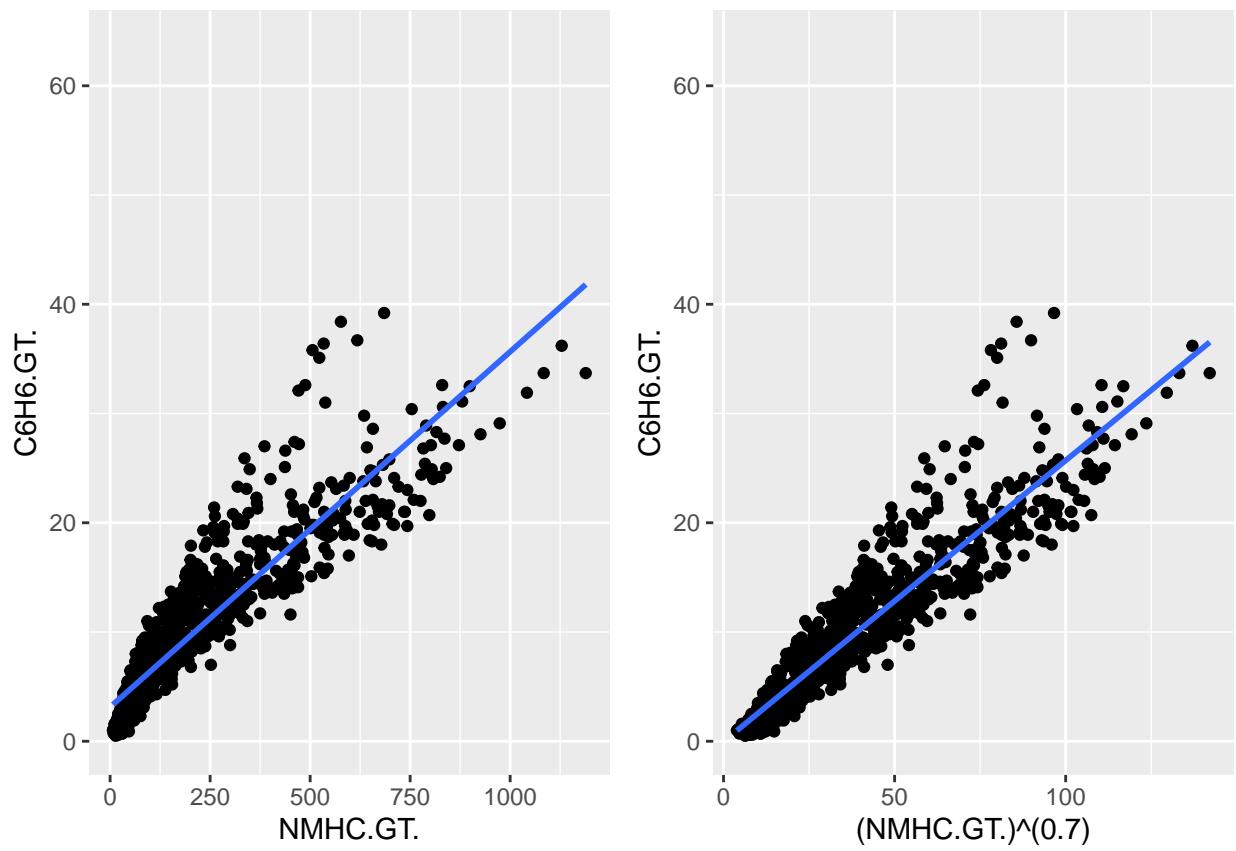
```

plot <- function(equation) {
  ggplot(air_quality, aes_string(x=equation, y='C6H6.GT.')) +
    geom_point() +
    geom_smooth(method='lm', se=F)
}

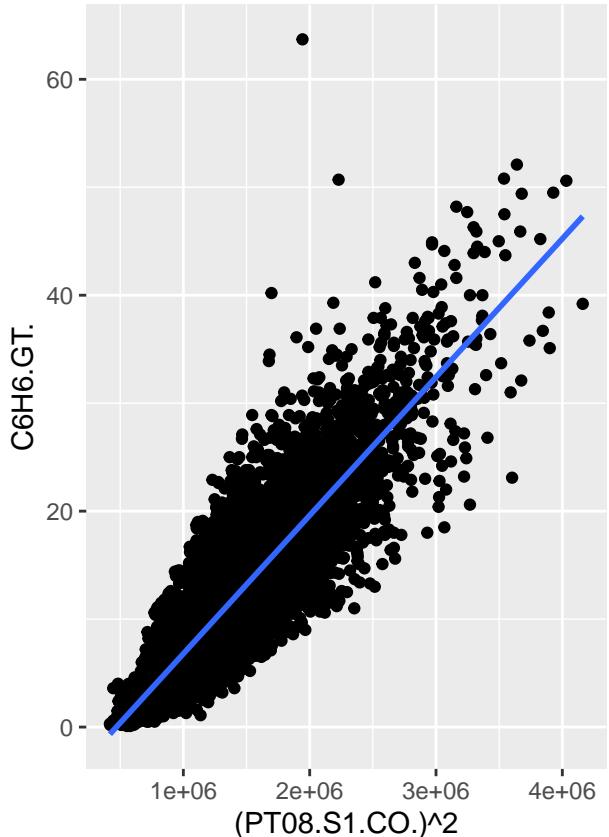
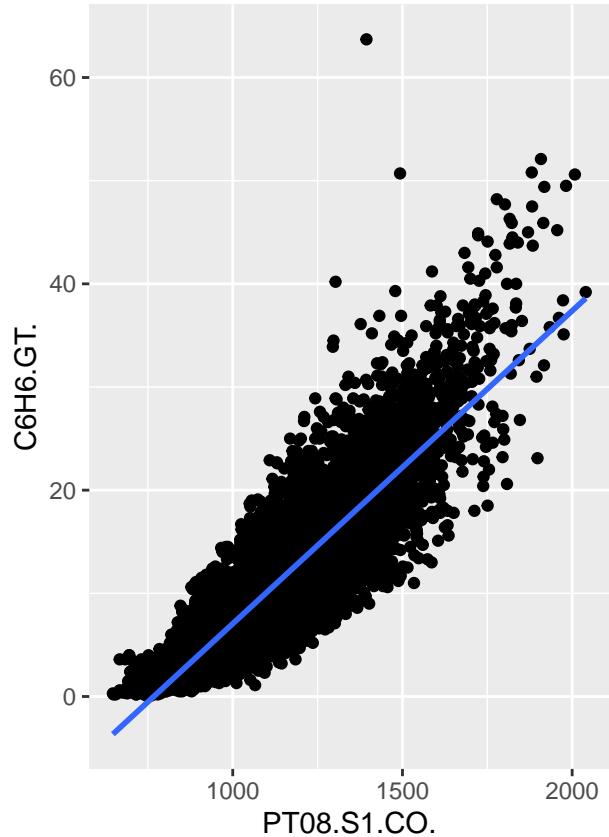
equations <- c('NMHC.GT.', '(NMHC.GT.)^(0.7)',
              'PT08.S1.CO.', '(PT08.S1.CO.)^2',
              'PT08.S2.NMHC.', '(PT08.S2.NMHC.)^2',
              'PT08.S3.NOx.', '1/(PT08.S3.NOx.)^2',
              'PT08.S4.NO2.', '(PT08.S4.NO2.)^(2.5)',
              'PT08.S5.O3.', '(PT08.S5.O3.)^(1.5)')
plots <- lapply(equations, plot)
ggarrange(plotlist=plots, ncol=2)

```

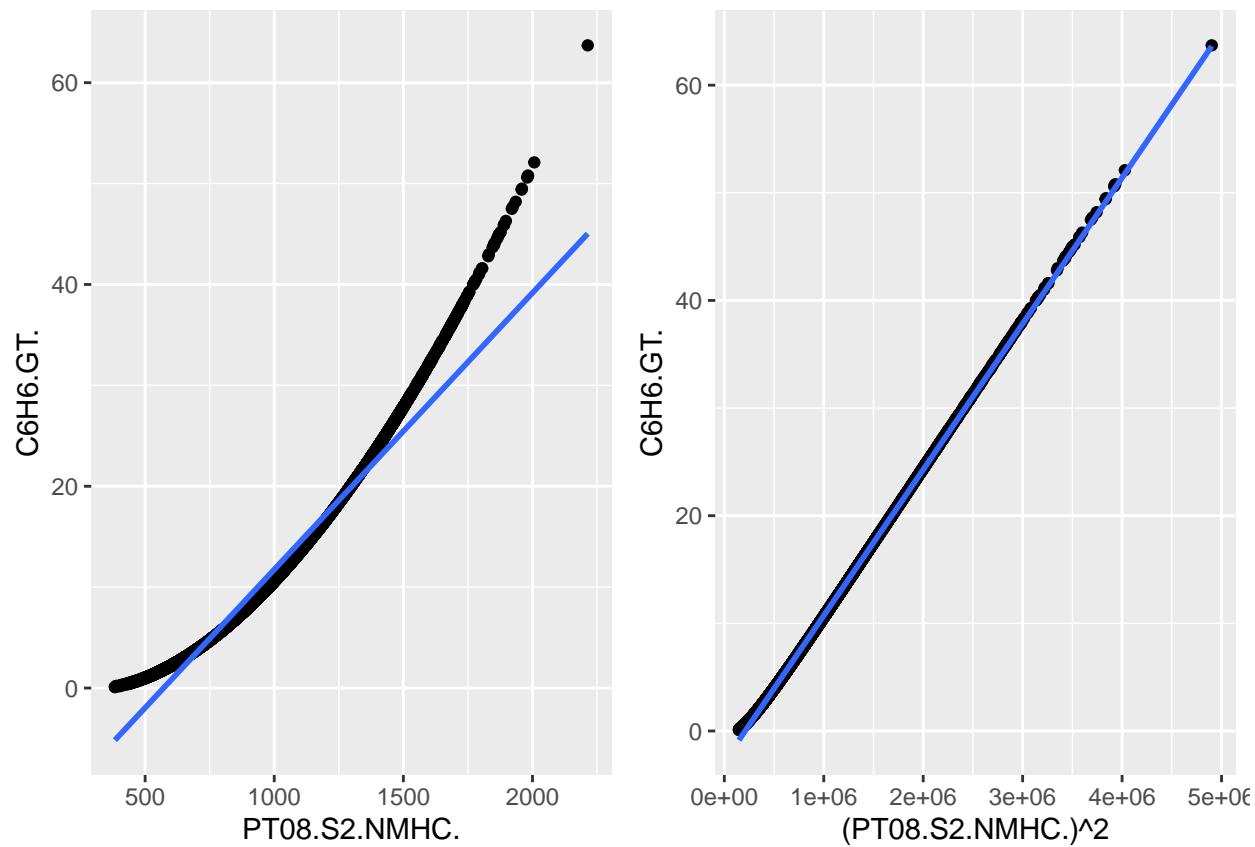
\$`1`



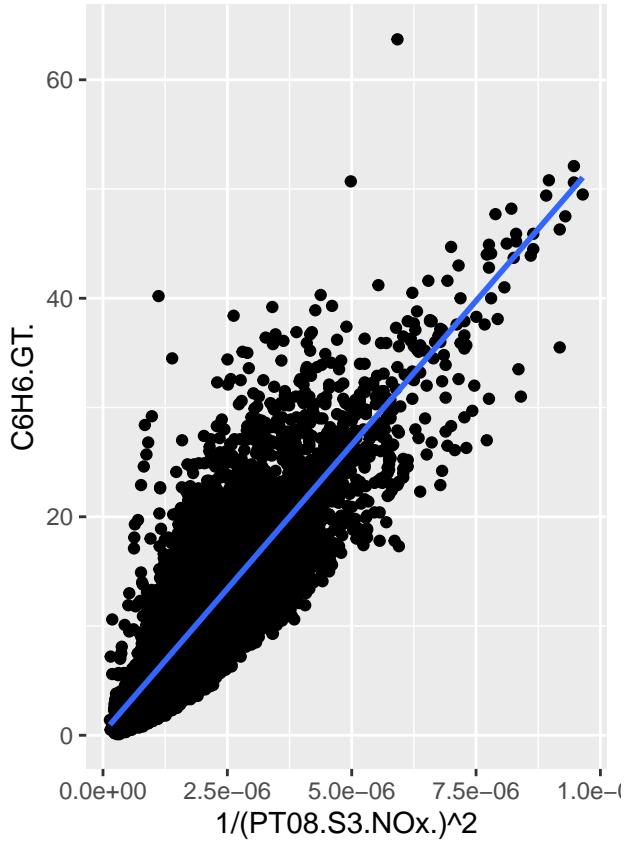
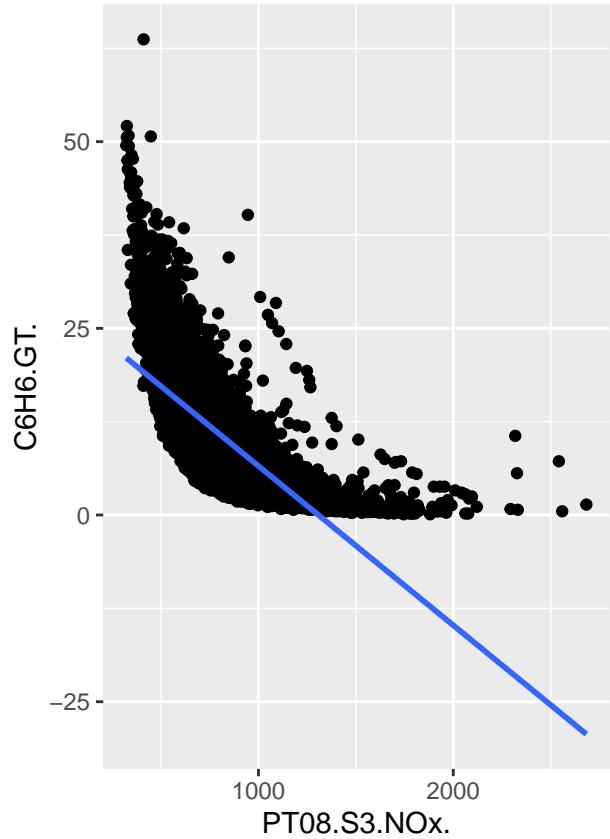
```
##  
## $^2`
```



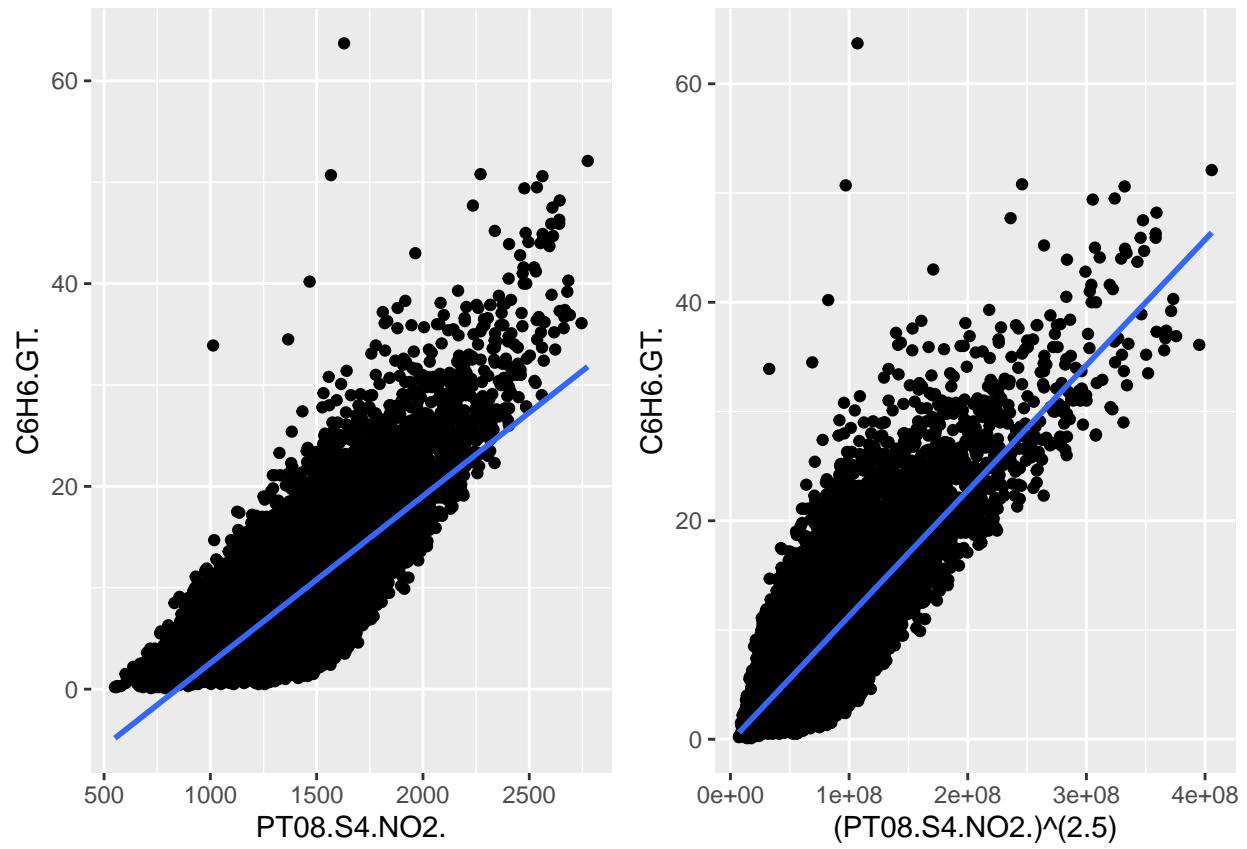
```
##  
## $`3`
```



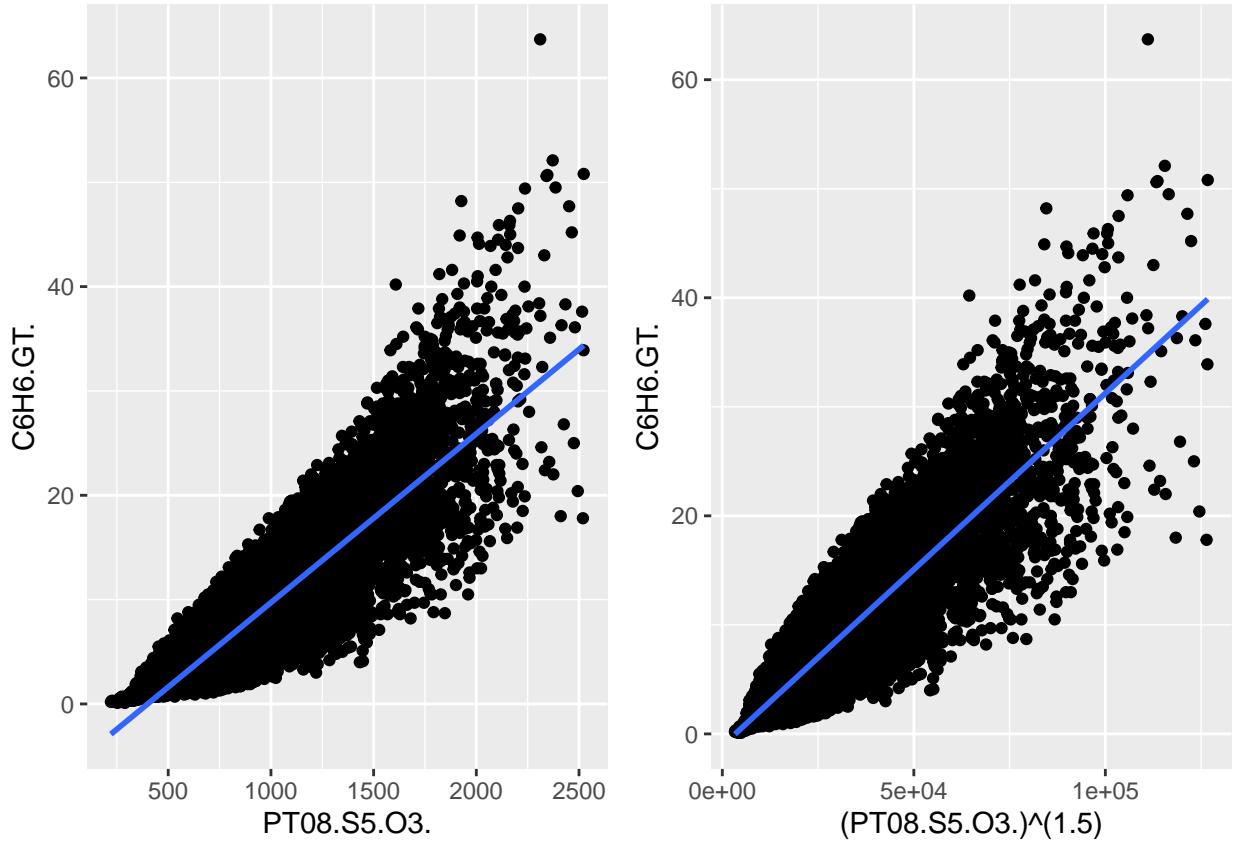
```
##  
## $^4`
```



```
##  
## $`5`
```



```
##  
## $`6`
```



```
##  
## attr(,"class")  
## [1] "list"      "ggarrange"  
  
airq_1 <- air_quality  
airq_1$`NMHC.GT.^0.7` <- air_quality$NMHC.GT.^0.7  
airq_1$`PT08.S1.CO.^2` <- air_quality$PT08.S1.CO.^2  
airq_1$`PT08.S2.NMHC.^2` <- air_quality$PT08.S2.NMHC.^2  
airq_1$`PT08.S3.NOx.^2` <- 1/air_quality$PT08.S3.NOx.^2  
airq_1$`PT08.S4.NO2.^2.5` <- air_quality$PT08.S4.NO2.^2.5  
airq_1$`PT08.S5.O3.^1.5` <- air_quality$PT08.S5.O3.^1.5
```

Building linear models

```
benzene_lm_summary <- function(predictor) {  
  airq_1 %>%  
    lm(data = ., C6H6.GT. ~ predictor) %>%  
    summary()  
}  
  
lapply(airq_1[, !names(airq_1) %in% c('Date', 'Time', 'C6H6.GT.',  
                                     'T', 'RH', 'AH',  
                                     'NMHC.GT.', 'PT08.S1.CO.', 'PT08.S2.NMHC.'),
```

```

'PT08.S3.NOx.', 'PT08.S4.NO2.', 'PT08.S5.03.')),  

benzene_lm_summary)

## $CO.GT.  

##  

## Call:  

## lm(formula = C6H6.GT. ~ predictor, data = .)  

##  

## Residuals:  

##      Min       1Q   Median       3Q      Max  

## -15.206  -1.532  -0.161   1.455  48.744  

##  

## Coefficients:  

##                 Estimate Std. Error t value Pr(>|t|)  

## (Intercept) 0.003968  0.056656   0.07   0.944  

## predictor   4.823080  0.022055 218.68 <2e-16 ***  

## ---  

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  

##  

## Residual standard error: 2.715 on 7342 degrees of freedom  

##   (2013 observations deleted due to missingness)  

## Multiple R-squared:  0.8669, Adjusted R-squared:  0.8669  

## F-statistic: 4.782e+04 on 1 and 7342 DF,  p-value: < 2.2e-16  

##  

##  

## $NOx.GT.  

##  

## Call:  

## lm(formula = C6H6.GT. ~ predictor, data = .)  

##  

## Residuals:  

##      Min       1Q   Median       3Q      Max  

## -12.926  -3.974  -1.115   2.840  38.846  

##  

## Coefficients:  

##                 Estimate Std. Error t value Pr(>|t|)  

## (Intercept) 4.0861758  0.0927714   44.05 <2e-16 ***  

## predictor   0.0259280  0.0002916   88.92 <2e-16 ***  

## ---  

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  

##  

## Residual standard error: 5.174 on 7394 degrees of freedom  

##   (1961 observations deleted due to missingness)  

## Multiple R-squared:  0.5167, Adjusted R-squared:  0.5167  

## F-statistic:  7906 on 1 and 7394 DF,  p-value: < 2.2e-16  

##  

##  

## $NO2.GT.  

##  

## Call:  

## lm(formula = C6H6.GT. ~ predictor, data = .)  

##  

## Residuals:

```

```

##      Min     1Q Median     3Q    Max
## -18.063 -3.454 -0.595  2.468 43.171
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.403719  0.174719 -2.311  0.0209 *
## predictor    0.096021  0.001434 66.960 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.872 on 7391 degrees of freedom
##   (1964 observations deleted due to missingness)
## Multiple R-squared:  0.3776, Adjusted R-squared:  0.3775
## F-statistic:  4484 on 1 and 7391 DF,  p-value: < 2.2e-16
##
##
## $`NMHC.GT.^0.7`
##
## Call:
## lm(formula = C6H6.GT. ~ predictor, data = .)
##
## Residuals:
##      Min     1Q Median     3Q    Max
## -6.929 -1.583 -0.626  1.144 16.387
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02808   0.17515  -0.16   0.873
## predictor    0.25728   0.00365  70.48 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.897 on 885 degrees of freedom
##   (8470 observations deleted due to missingness)
## Multiple R-squared:  0.8488, Adjusted R-squared:  0.8486
## F-statistic:  4968 on 1 and 885 DF,  p-value: < 2.2e-16
##
##
## $`PT08.S1.CO.^2`
##
## Call:
## lm(formula = C6H6.GT. ~ predictor, data = .)
##
## Residuals:
##      Min     1Q Median     3Q    Max
## -17.039 -1.971 -0.225  1.694 44.821
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.020e+00  9.456e-02 -63.67 <2e-16 ***
## predictor    1.281e-05  6.958e-08 184.14 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

```

## Residual standard error: 3.41 on 8989 degrees of freedom
##   (366 observations deleted due to missingness)
## Multiple R-squared:  0.7905, Adjusted R-squared:  0.7904
## F-statistic: 3.391e+04 on 1 and 8989 DF, p-value: < 2.2e-16
##
##
## $`PT08.S2.NMHC.^2`
##
## Call:
## lm(formula = C6H6.GT. ~ predictor, data = .)
##
## Residuals:
##       Min     1Q   Median     3Q    Max
## -0.18511 -0.10417 -0.05005  0.06007  1.01125
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.842e+00  3.261e-03 -871.6  <2e-16 ***
## predictor    1.356e-05  2.964e-09  4574.6  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1544 on 8989 degrees of freedom
##   (366 observations deleted due to missingness)
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9996
## F-statistic: 2.093e+07 on 1 and 8989 DF, p-value: < 2.2e-16
##
##
## $`PT08.S3.NOx.^2`
##
## Call:
## lm(formula = C6H6.GT. ~ predictor, data = .)
##
## Residuals:
##      Min     1Q   Median     3Q    Max
## -14.283 -2.400  -0.897   1.808  34.077
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.345e-01  7.699e-02   3.046  0.00233 **
## predictor   5.270e+06  3.467e+04 152.000  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.943 on 8989 degrees of freedom
##   (366 observations deleted due to missingness)
## Multiple R-squared:  0.7199, Adjusted R-squared:  0.7199
## F-statistic: 2.31e+04 on 1 and 8989 DF, p-value: < 2.2e-16
##
##
## $`PT08.S4.NO2.^2.5`
##
## Call:
## lm(formula = C6H6.GT. ~ predictor, data = .)

```

```

##
## Residuals:
##      Min     1Q Median     3Q    Max
## -9.065 -3.127 -0.626  2.266 51.601
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.964e-01  9.163e-02 -2.143   0.0321 *
## predictor    1.148e-07  8.831e-10 129.996  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.39 on 8989 degrees of freedom
##   (366 observations deleted due to missingness)
## Multiple R-squared:  0.6528, Adjusted R-squared:  0.6527
## F-statistic: 1.69e+04 on 1 and 8989 DF,  p-value: < 2.2e-16
##
##
## $`PT08.S5.03.^1.5`
##
## Call:
## lm(formula = C6H6.GT. ~ predictor, data = .)
##
## Residuals:
##      Min     1Q Median     3Q    Max
## -21.9820 -2.0572 -0.1852  1.9298 28.8736
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.084e+00  7.654e-02 -14.16   <2e-16 ***
## predictor    3.232e-04  1.915e-06 168.77  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.649 on 8989 degrees of freedom
##   (366 observations deleted due to missingness)
## Multiple R-squared:  0.7601, Adjusted R-squared:  0.7601
## F-statistic: 2.848e+04 on 1 and 8989 DF,  p-value: < 2.2e-16

```

Selected predictors

PT08.S2(NMHC)² - titanium dioxide ($R^2 = 0.9996$), CO(GT) - carbon monoxide ($R^2 = 0.8669$), NMHC(GT)^{0.7} - nonmethane hydrocarbons ($R^2 = 0.8486$).

```

set.seed(42)
sample <- sample.int(n=nrow(airq_1), size=floor(.75*nrow(airq_1)))
train <- airq_1[sample, ]
test <- airq_1[-sample, ]

model_PT08S2 <- lm(data=train, C6H6.GT. ~ `PT08.S2.NMHC.^2`)
model_CO <- lm(data=train, C6H6.GT. ~ CO.GT.)
model_NMHC <- lm(data=train, C6H6.GT. ~ `NMHC.GT.^0.7`)

```

```

test$C6H6byPT08S2 <- predict(model_PT08S2, newdata=test)
test$C6H6byCO <- predict(model_CO, newdata=test)
test$C6H6byNMHC <- predict(model_NMHC, newdata=test)

```

Plots

```

plot <- function(lm_info) {
  ggplot() +
    geom_point(data=train, aes_string(x=lm_info[[1]], y='C6H6.GT.',
                                       color=shQuote('train'), shape=shQuote('train'))) +
    geom_point(data=test, aes_string(x=lm_info[[1]], y='C6H6.GT.',
                                      color=shQuote('test'), shape=shQuote('test'))) +
    geom_smooth(method='lm', se=F, data=train, aes_string(x=lm_info[[1]], y='C6H6.GT.'),
                color='black') +
    geom_point(data=test, aes_string(x=lm_info[[1]], y=lm_info[[2]],
                                     color=shQuote('predicted'),
                                     shape=shQuote('predicted'))) +
    scale_colour_manual(name="data", breaks=c('train', 'test', 'predicted'),
                        values=c(train="blue", test="red", predicted="green3")) +
    scale_shape_manual(name="data", breaks=c('train', 'test', 'predicted'),
                       values=c(train=1, test=2, predicted=20)) +
    ggtitle(paste("R-squared = ", lm_info[[3]]$adj.r.squared, '\n',
                  "p-value = ", lm_info[[3]]$coefficients[,4][[2]]))
}

# list(predictor, predicted, linear model summary)
PT08S2 <- list('PT08.S2.NMHC.^2', 'C6H6byPT08S2', summary(model_PT08S2))
CO      <- list('CO.GT.', 'C6H6byCO', summary(model_CO))
NMHC   <- list('NMHC.GT.^0.7', 'C6H6byNMHC', summary(model_NMHC))

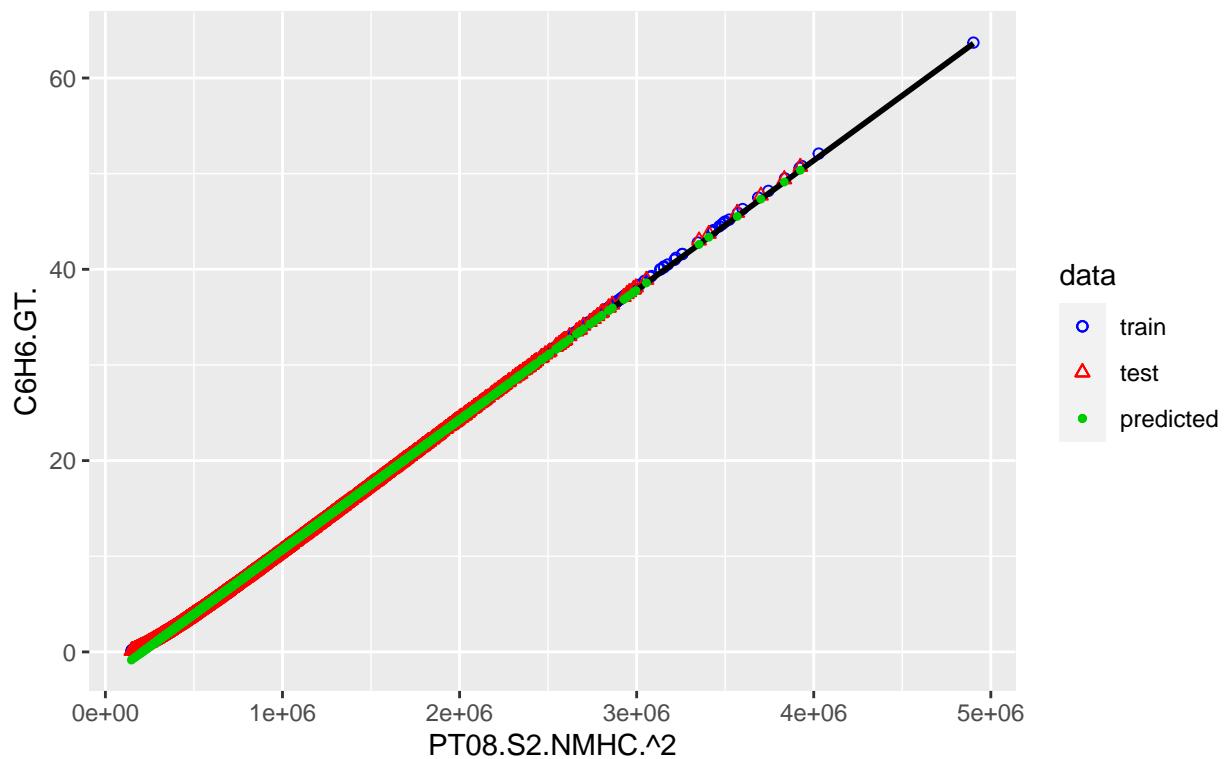
info <- list(PT08S2 = PT08S2, CO = CO, NMHC = NMHC)
lapply(info, plot)

## $PT08S2

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

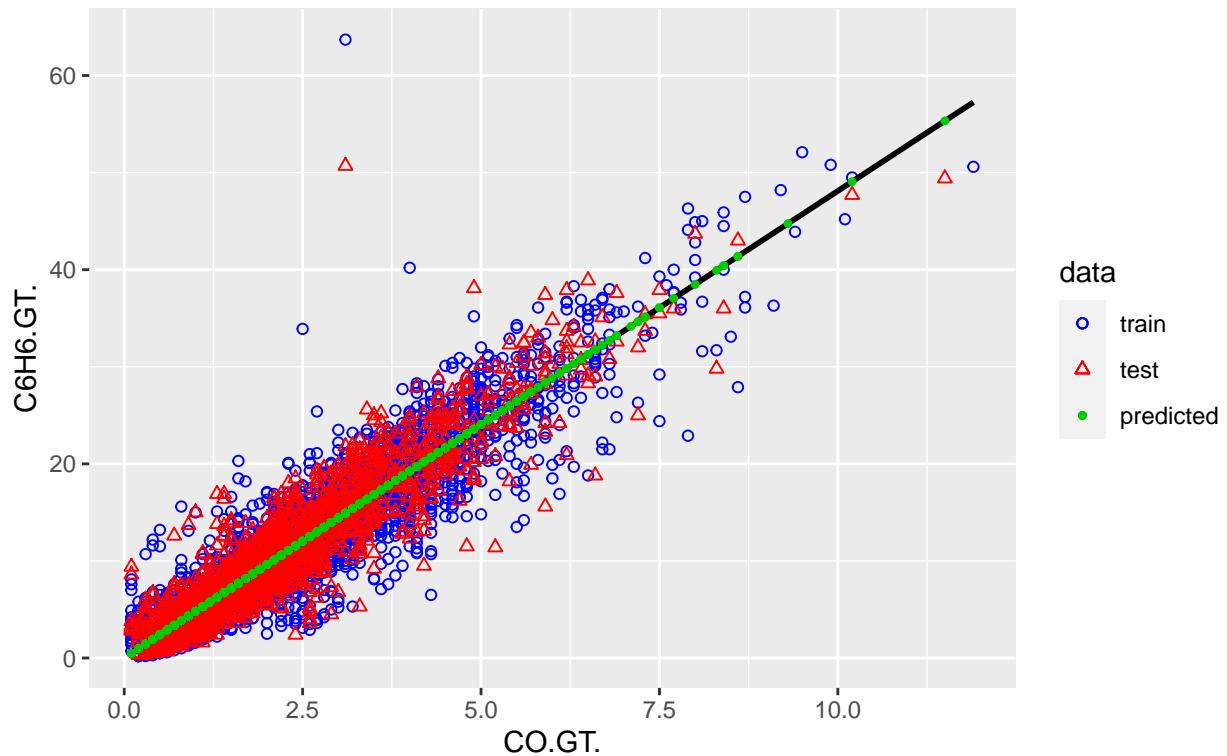
```

R-squared = 0.999555520216006
p-value = 0



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

R-squared = 0.867795000862623
p-value = 0



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

R-squared = 0.841035698748048
p-value = 3.48548269311541e-267

