HW2.1_Mary_Futey

Mary Futey

4/3/2020

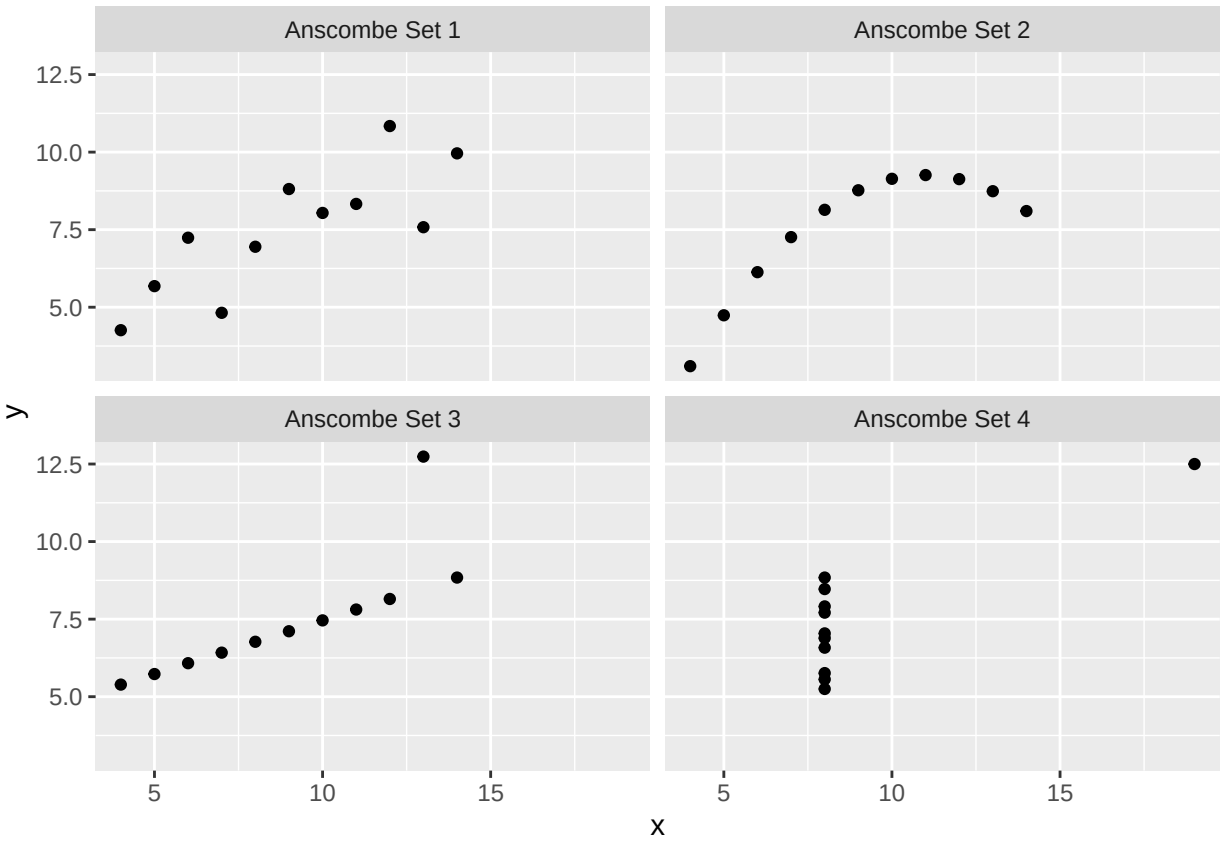
Part 1: Anscombe's dataset

*Scatter plot facetted by set

```
knitr::kable(anscombe)
```

x1	x2	x3	x4	y1	y2	y3	y4
10	10	10	8	8.04	9.14	7.46	6.58
8	8	8	8	6.95	8.14	6.77	5.76
13	13	13	8	7.58	8.74	12.74	7.71
9	9	9	8	8.81	8.77	7.11	8.84
11	11	11	8	8.33	9.26	7.81	8.47
14	14	14	8	9.96	8.10	8.84	7.04
6	6	6	8	7.24	6.13	6.08	5.25
4	4	4	19	4.26	3.10	5.39	12.50
12	12	12	8	10.84	9.13	8.15	5.56
7	7	7	8	4.82	7.26	6.42	7.91
5	5	5	8	5.68	4.74	5.73	6.89

```
anscombe.1 <- data.frame(x = anscombe[["x1"]],
                         y = anscombe[["y1"]],
                         Set = "Anscombe Set 1")
anscombe.2 <- data.frame(x =
                           anscombe[["x2"]],
                         y = anscombe[["y2"]],
                         Set = "Anscombe Set 2")
anscombe.3 <- data.frame(x = anscombe[["x3"]],
                         y = anscombe[["y3"]],
                         Set = "Anscombe Set 3")
anscombe.4 <- data.frame(x = anscombe[["x4"]],
                         y = anscombe[["y4"]],
                         Set = "Anscombe Set 4")
anscombe.data <- rbind(anscombe.1, anscombe.2, anscombe.3, anscombe.4)

ans_facet <- ggplot(anscombe.data,
                    aes(x = x,
                        y = y))+
  geom_point(color = "black")+
  facet_wrap(~Set, ncol = 2)

ans_facet
```



*Summary calculation (mean, sd) grouped by set

```
aggregate(cbind(x, y) ~ Set, anscombe.data, mean)
```

```
##              Set x        y
## 1 Anscombe Set 1 9 7.500909
## 2 Anscombe Set 2 9 7.500909
## 3 Anscombe Set 3 9 7.500000
## 4 Anscombe Set 4 9 7.500909
```

```
aggregate(cbind(x, y) ~ Set, anscombe.data, sd)
```

```
##              Set        x        y
## 1 Anscombe Set 1 3.316625 2.031568
## 2 Anscombe Set 2 3.316625 2.031657
## 3 Anscombe Set 3 3.316625 2.030424
## 4 Anscombe Set 4 3.316625 2.030579
```

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

```
correlation <- function(data) {
  a <- data.frame(pearson = cor.test(data$x, data$y, method = "pearson")$statistic)
  b <- data.frame(p_pearson = cor.test(data$x, data$y, method = "pearson")$p.value)
  c <- data.frame(kendall = cor.test(data$x, data$y, method = "kendall")$statistic)
  d <- data.frame(p_kendall = cor.test(data$x, data$y, method = "kendall")$p.value)
  e <- data.frame(spearman = cor.test(data$x, data$y, method = "spearman")$statistic)
  f <- data.frame(p_spearman = cor.test(data$x, data$y, method = "spearman")$p.value)

  return(list(a, b, c, d, e, f))
```

```
}

res <- correlation(anscombe.data)
```

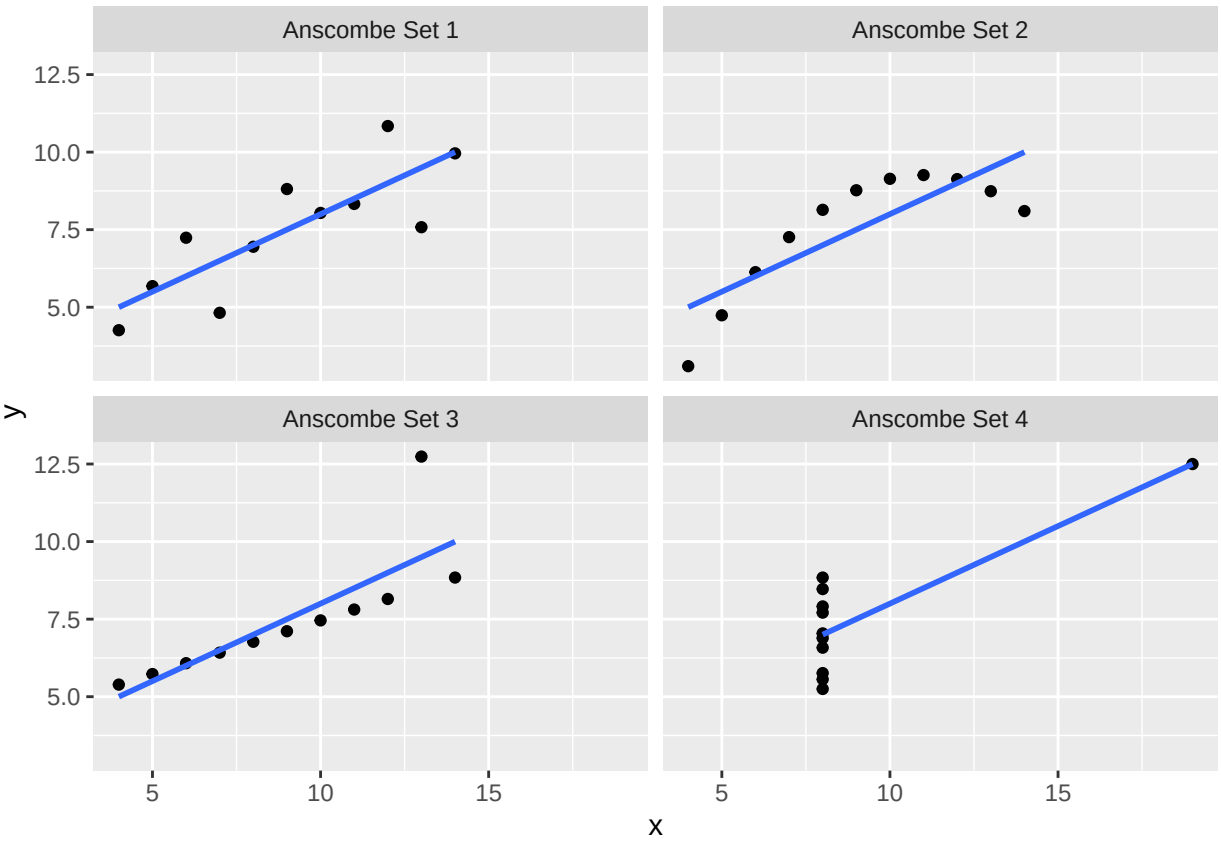
```
## Warning in cor.test.default(data$x, data$y, method = "kendall"): Cannot compute
## exact p-value with ties

## Warning in cor.test.default(data$x, data$y, method = "kendall"): Cannot compute
## exact p-value with ties

## Warning in cor.test.default(data$x, data$y, method = "spearman"): Cannot compute
## exact p-value with ties

## Warning in cor.test.default(data$x, data$y, method = "spearman"): Cannot compute
## exact p-value with ties
```

```
ldply(res)
```

```
##    pearson    p_pearson kendall    p_kendall spearman   p_spearman
## 1 9.160764           NA      NA           NA       NA           NA
## 2       NA 1.436505e-11      NA           NA       NA           NA
## 3       NA           NA 6.05286           NA       NA           NA
## 4       NA           NA      NA 1.422967e-09       NA           NA
## 5       NA           NA      NA           NA 2598.395           NA
## 6       NA           NA      NA           NA       NA 1.360916e-11
```

*Add geom_smooth()totheplot

```
ans_smooth <- ans_facet +
  geom_smooth(formula = y ~ x, method = "lm", se = FALSE, data = anscombe.data)
ans_smooth
```



Air Quality dataset

- Explore data set, clean if needed: convert date and time to factor, check for NAs, switch comma to decimal
- Explore each variable independently: remove "-200" values as they are not possible / erroneous

```
airq <- read.csv("/Users/maryfutey/desktop/AirQualityUCI/AirQualityUCI.csv",
                 header = TRUE,
                 dec=",")
head(airq)
```

```
##         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
```

```
airq_long <- gather(airq, key="measurement", value="value", -c(Date,Time))
```

```r
airq_long$Date <- as.factor(airq_long$Date)
airq_long$Time <- as.factor(airq_long$Time)
airq_long$measurement <- as.factor(airq_long$measurement)

airq_fil <- airq_long %>%
  filter_all(all_vars(. != -200))

head(airq_fil)
```

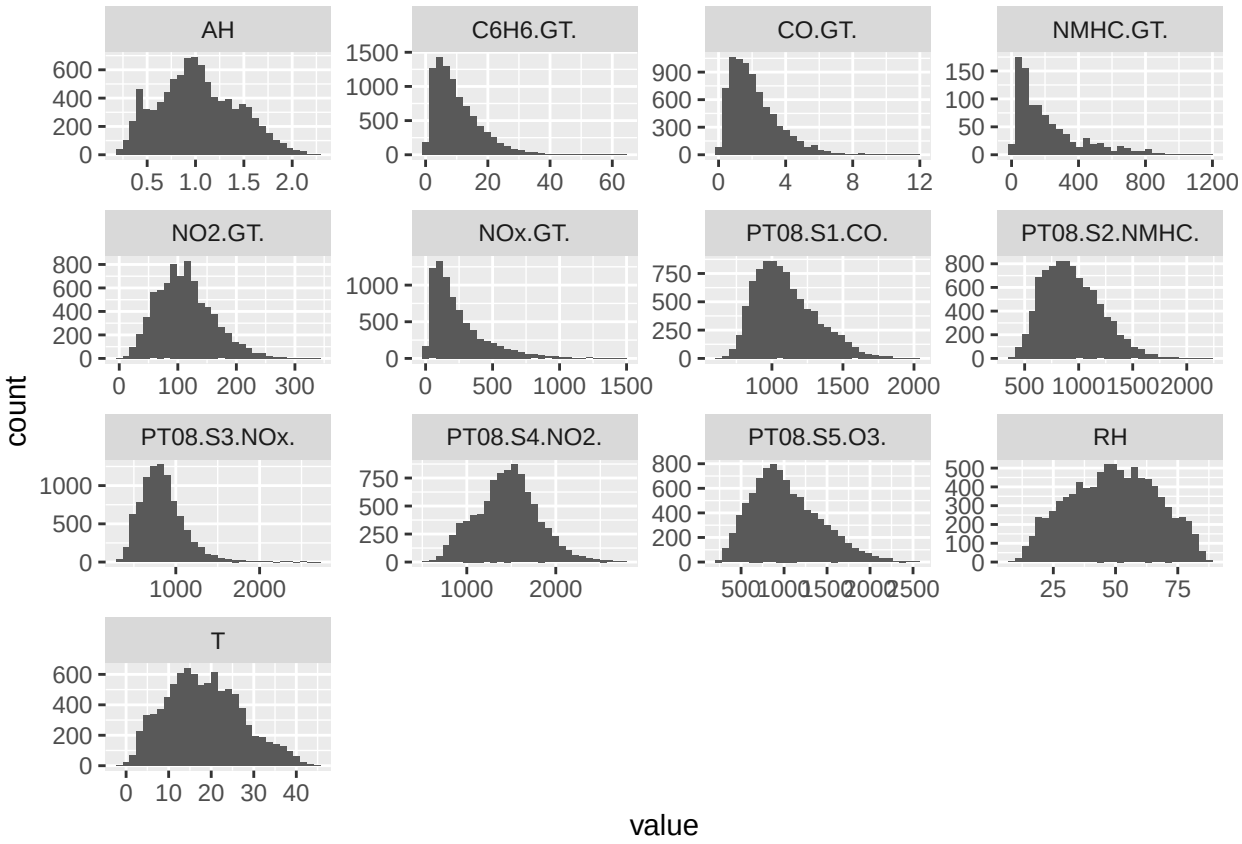
```
##         Date     Time measurement value
## 1 10/03/2004 18.00.00      CO.GT.   2.6
## 2 10/03/2004 19.00.00      CO.GT.   2.0
## 3 10/03/2004 20.00.00      CO.GT.   2.2
## 4 10/03/2004 21.00.00      CO.GT.   2.2
## 5 10/03/2004 22.00.00      CO.GT.   1.6
## 6 10/03/2004 23.00.00      CO.GT.   1.2
```

```r
colSums(is.na(airq_fil))
```

```
##        Date        Time measurement       value
##           0           0           0           0
```

```r
p1 <- airq_fil %>% ggplot(aes(x = value)) +
  geom_histogram() +
  facet_wrap(~measurement, scales = "free")

p1
```

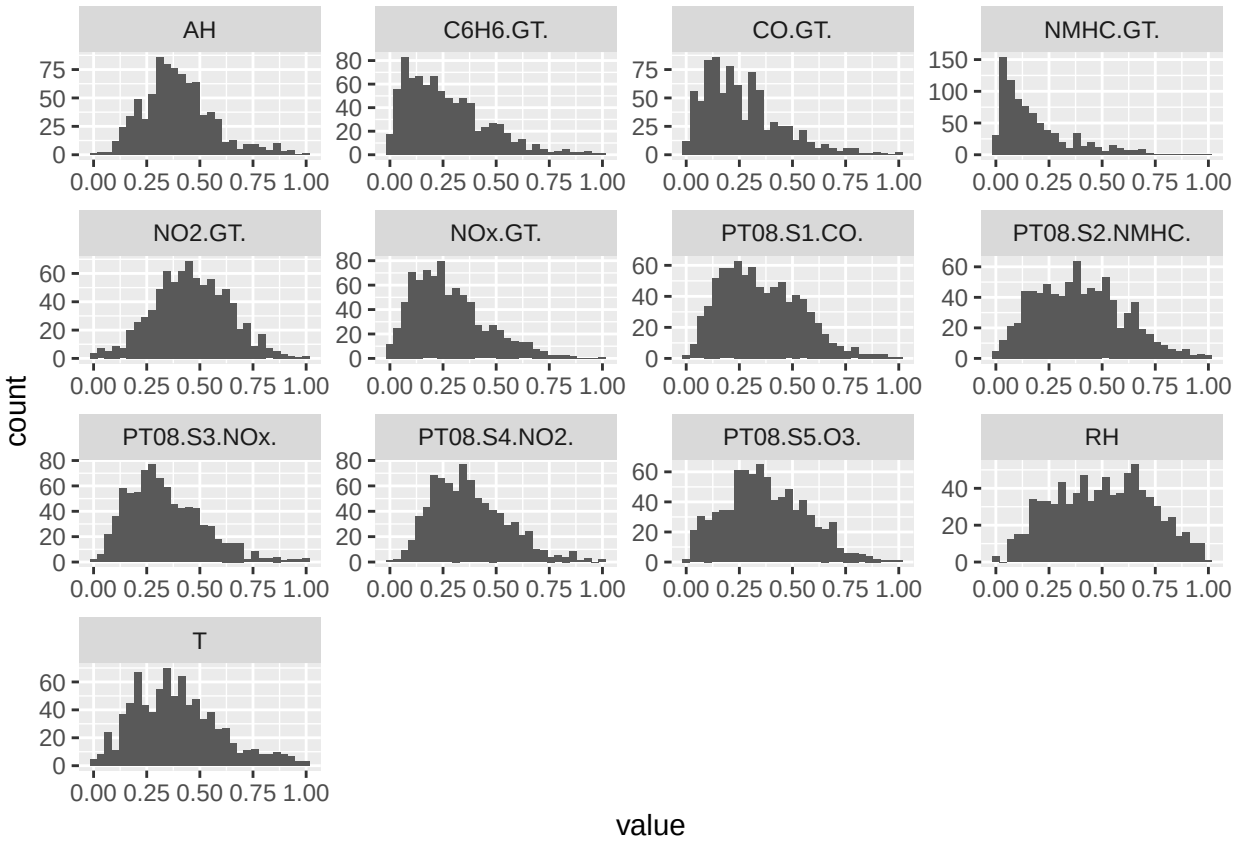
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- Need to normalize data

```
airq_wide <- spread(airq_fil, key = "measurement" ,value = "value")
airq_wide <- na.omit(airq_wide)

norm <- function(x) {
    (x - min(x)) / (max(x) - min(x))
  }

airq_norm <- as.data.frame(lapply(airq_wide[3:15], norm))
airq_norm$Date <- airq_wide$Date
airq_norm$Time <- airq_wide$Time


airq_norm_long <- gather(airq_norm, key = "measurement", value = "value", -c(Date, Time))

p2 <- airq_norm_long %>%  ggplot(aes(x = value)) +
  geom_histogram() +
  facet_wrap(~measurement, scales = "free")

p2
```

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



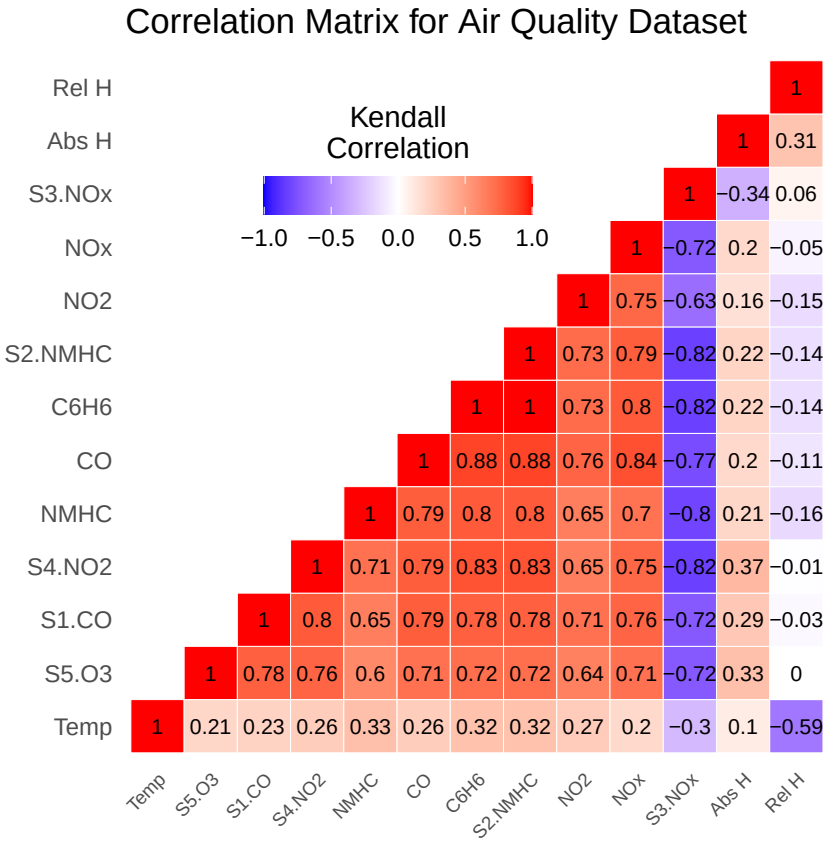
- Cross correlations

```
airq_sub <- airq_norm[, 1:13]
colnames(airq_sub) <- c(
  "Abs H",
  "C6H6",
  "CO",
  "NMHC",
  "NO2",
  "NOx",
  "S1.CO",
  "S2.NMHC",
  "S3.NOx",
  "S4.NO2",
  "S5.O3",
  "Rel H",
  "Temp"
)

airq_cor <- round(cor(airq_sub, method = "kendall"),2)

#reorder, create upper triangle
reorder_airq_cor <- function(airq_cor){
  # Use correlation between variables as distance
  dd <- as.dist((1-airq_cor)/2)
  hc <- hclust(dd)
  airq_cor <-airq_cor[hc$order, hc$order]
}
```

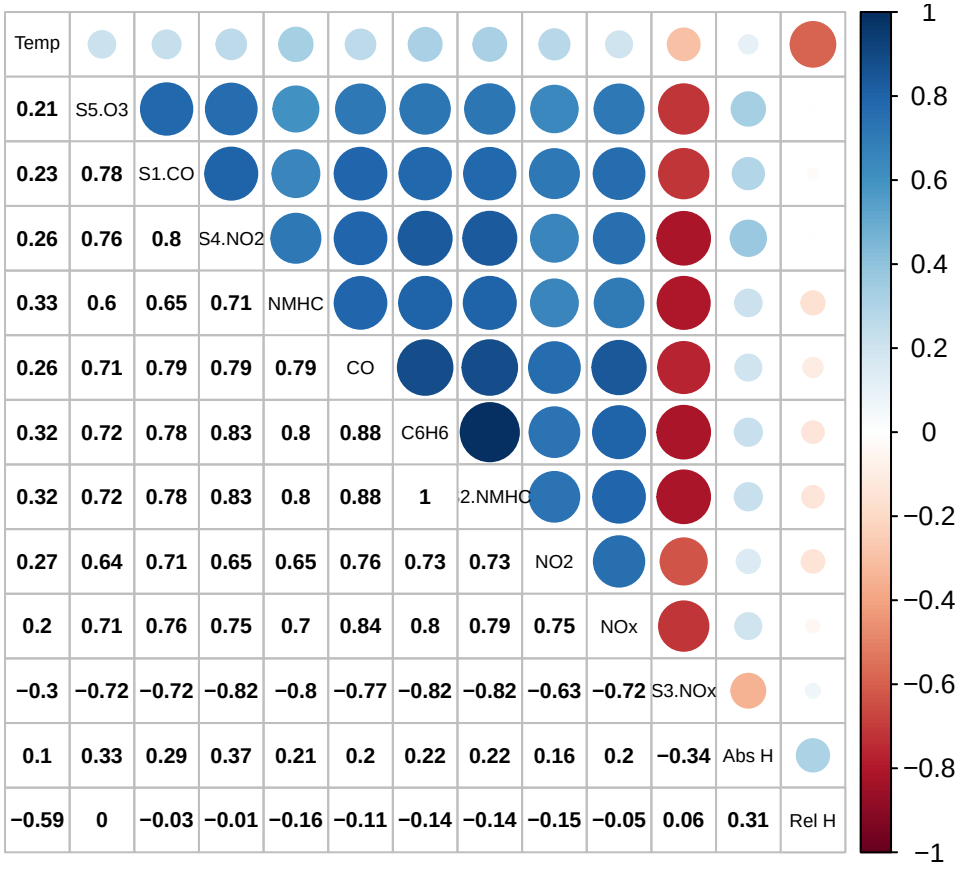
```r
get_upper_tri <- function(airq_cor){
  airq_cor[lower.tri(airq_cor)]<- NA
  return(airq_cor)
}

airq_cor <- reorder_airq_cor(airq_cor)
upper_tri <- get_upper_tri(airq_cor)
melt_uppertri <- melt(upper_tri, na.rm = TRUE)
```

```
## Warning in melt(upper_tri, na.rm = TRUE): The melt generic in data.table has
## been passed a matrix 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(upper_tri). In the next version, this warning will become an
## error.
```

```r
p3 <- ggplot(melt_uppertri,
             aes(Var2, Var1,
                 fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Kendall\nCorrelation") +
  ggtitle("Correlation Matrix for Air Quality Dataset") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 7, hjust = 1)) +
  coord_fixed() +
  geom_text(aes(Var2,
                Var1,
                label = value),
            color = "black", size = 2.8) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.ticks = element_blank(),
    legend.justification = c(1, 0),
    legend.position = c(0.6, 0.7),
    legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                               title.position = "top", title.hjust = 0.5))

p3
```




```
p4 <- corrplot.mixed(airq_cor,
                     tl.col = "black", tl.cex = 0.6,
                     lower.col = "black", number.cex = 0.7)
```




```
p4
```

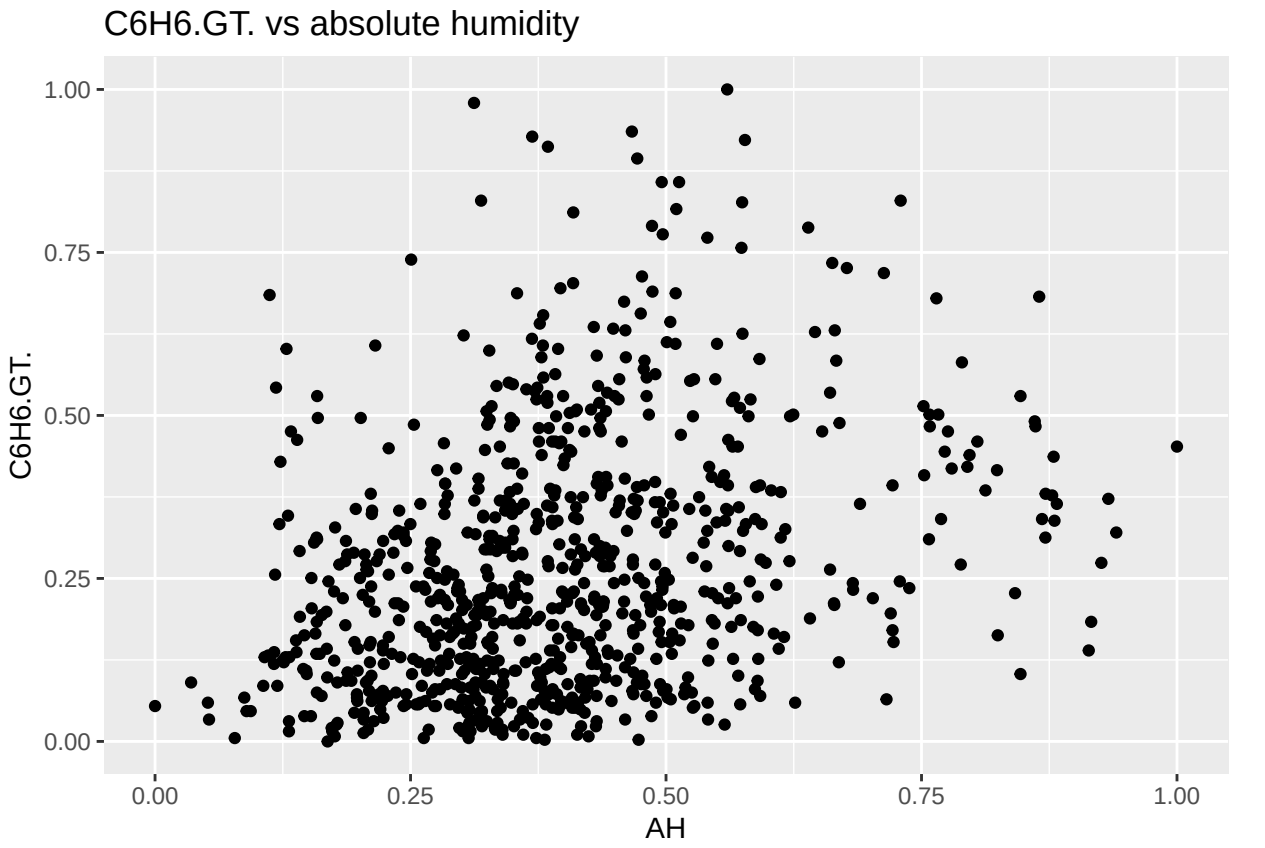
```
##           Temp S5.O3 S1.CO S4.NO2  NMHC    CO  C6H6 S2.NMHC   NO2   NOx S3.NOx
## Temp      1.00  0.21  0.23   0.26  0.33  0.26  0.32    0.32  0.27  0.20  -0.30
## S5.O3     0.21  1.00  0.78   0.76  0.60  0.71  0.72    0.72  0.64  0.71  -0.72
## S1.CO     0.23  0.78  1.00   0.80  0.65  0.79  0.78    0.78  0.71  0.76  -0.72
## S4.NO2    0.26  0.76  0.80   1.00  0.71  0.79  0.83    0.83  0.65  0.75  -0.82
## NMHC      0.33  0.60  0.65   0.71  1.00  0.79  0.80    0.80  0.65  0.70  -0.80
## CO        0.26  0.71  0.79   0.79  0.79  1.00  0.88    0.88  0.76  0.84  -0.77
## C6H6      0.32  0.72  0.78   0.83  0.80  0.88  1.00    1.00  0.73  0.80  -0.82
## S2.NMHC   0.32  0.72  0.78   0.83  0.80  0.88  1.00    1.00  0.73  0.79  -0.82
## NO2       0.27  0.64  0.71   0.65  0.65  0.76  0.73    0.73  1.00  0.75  -0.63
## NOx       0.20  0.71  0.76   0.75  0.70  0.84  0.80    0.79  0.75  1.00  -0.72
## S3.NOx   -0.30 -0.72 -0.72  -0.82 -0.80 -0.77 -0.82   -0.82 -0.63 -0.72   1.00
## Abs H     0.10  0.33  0.29   0.37  0.21  0.20  0.22    0.22  0.16  0.20  -0.34
## Rel H    -0.59  0.00 -0.03  -0.01 -0.16 -0.11 -0.14   -0.14 -0.15 -0.05   0.06
##           Abs H Rel H
## Temp       0.10 -0.59
## S5.O3      0.33  0.00
## S1.CO      0.29 -0.03
## S4.NO2     0.37 -0.01
## NMHC       0.21 -0.16
## CO         0.20 -0.11
## C6H6       0.22 -0.14
## S2.NMHC    0.22 -0.14
## NO2        0.16 -0.15
## NOx        0.20 -0.05
```

```
## S3.NOx  -0.34  0.06
## Abs H    1.00  0.31
## Rel H    0.31  1.00
```

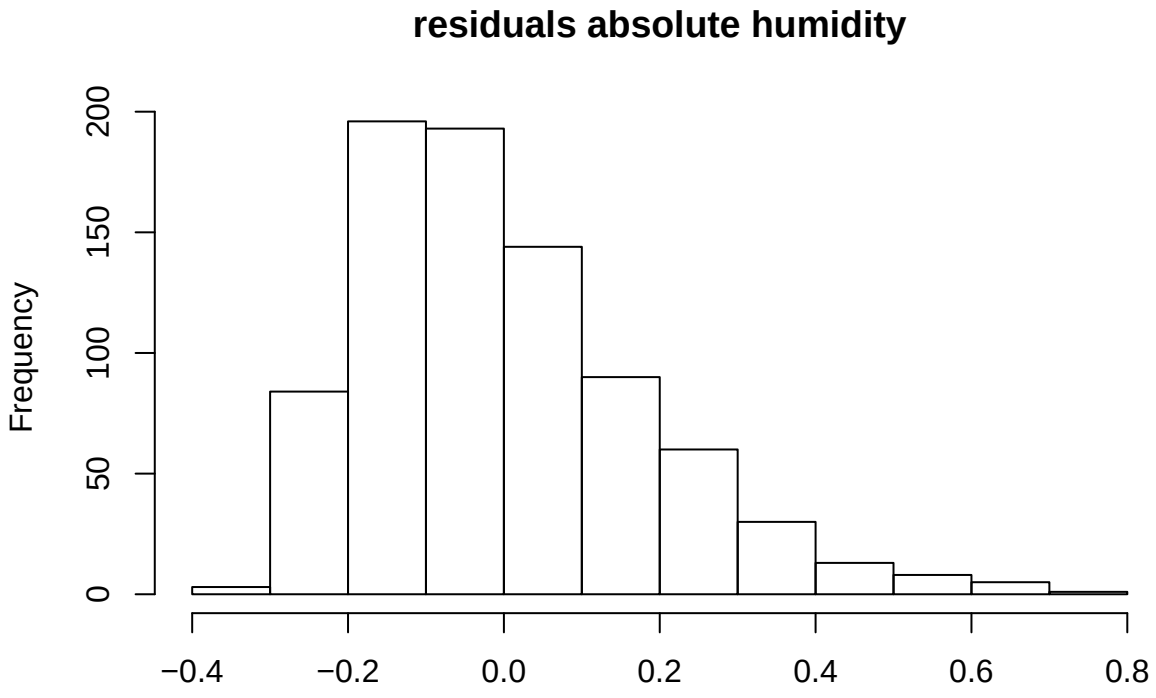
- Build simple linear models with each predictor, check assumptions, response C6H6

#Absolute humidity - data not linear

```
ggplot(airq_norm, aes(x = AH,
                      y = C6H6.GT.)) +
  geom_point() +
  ggtitle("C6H6.GT. vs absolute humidity")
```




```
AH <- lm(C6H6.GT. ~ AH, airq_norm)
residuals(AH) %>% hist(main = "residuals absolute humidity")
```



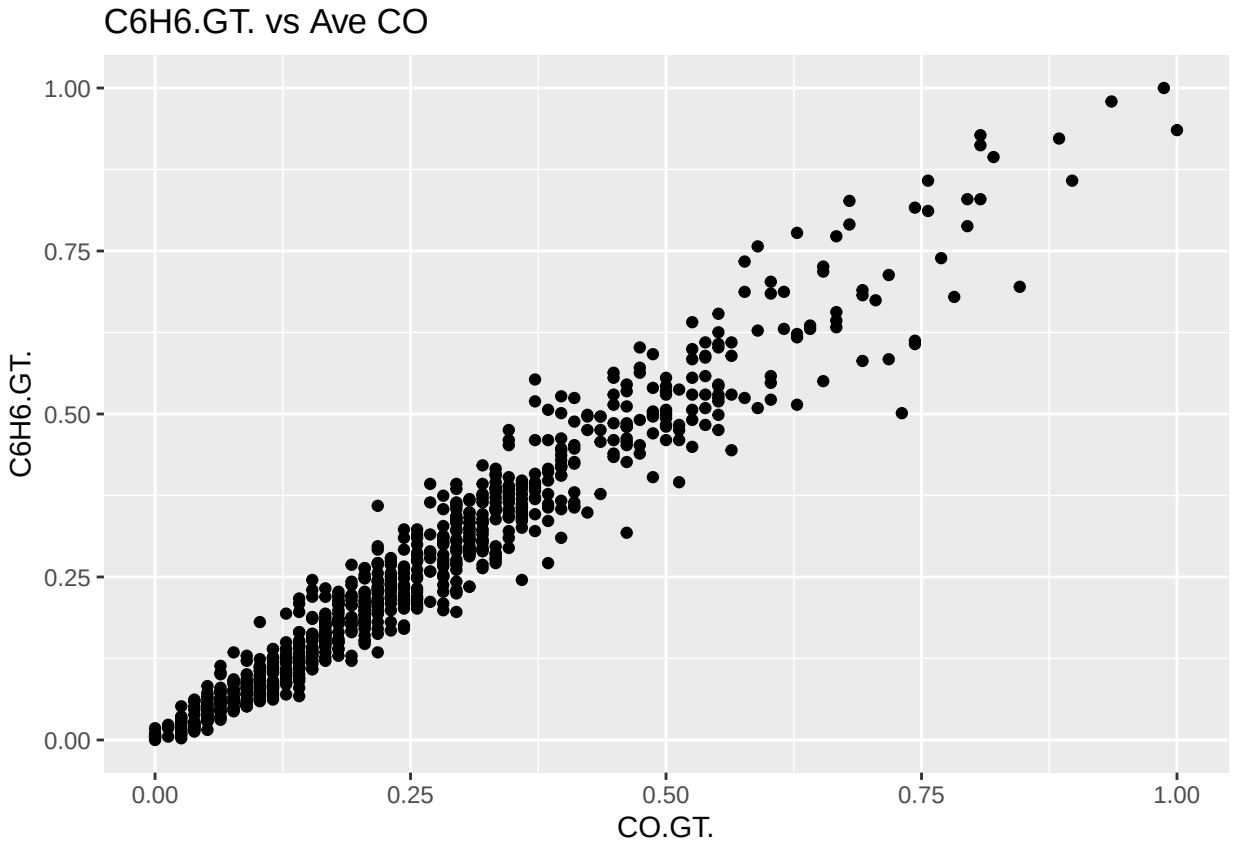
.

```
summary(AH)
```

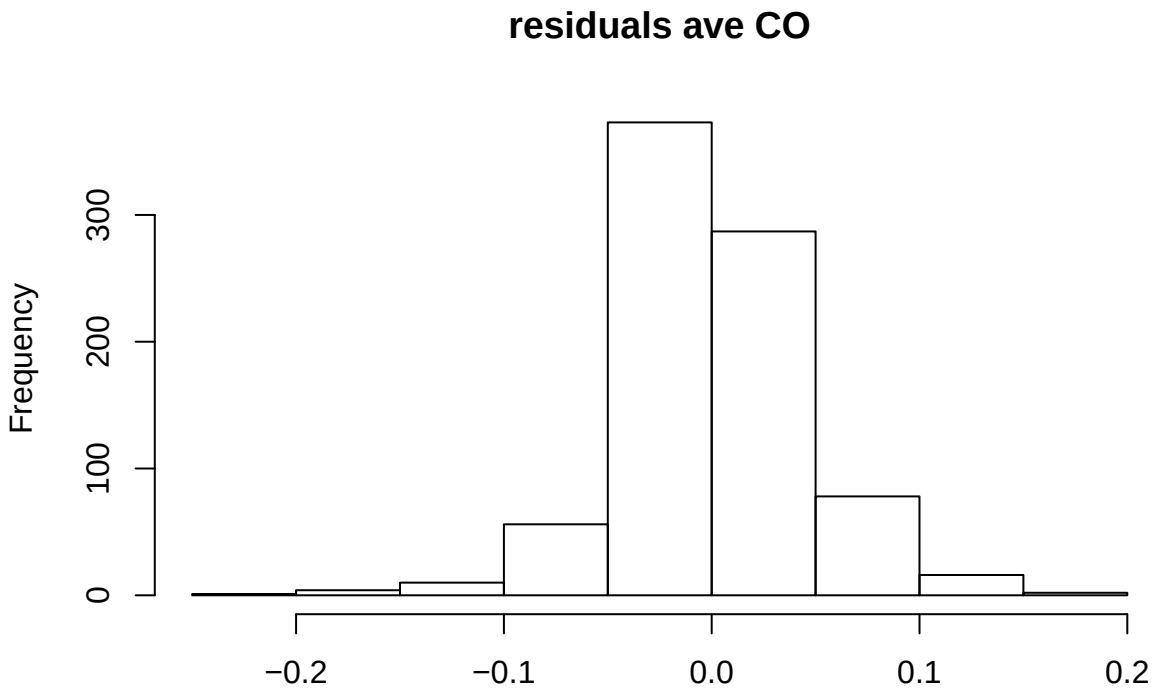
```
##
## Call:
## lm(formula = C6H6.GT. ~ AH, data = airq_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32613 -0.13882 -0.03953  0.09949  0.74474
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.12084    0.01651   7.318 5.99e-13 ***
## AH           0.36445    0.03845   9.480  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1821 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
```

#Ave CO (CO.GT.)

```
ggplot(airq_norm, aes(x = CO.GT.,
                      y = C6H6.GT.)) +
  geom_point() +
  ggtitle("C6H6.GT. vs Ave CO")
```




```
CO <- lm(C6H6.GT. ~ CO.GT., airq_norm)
residuals(CO) %>% hist(main = "residuals ave CO")
```



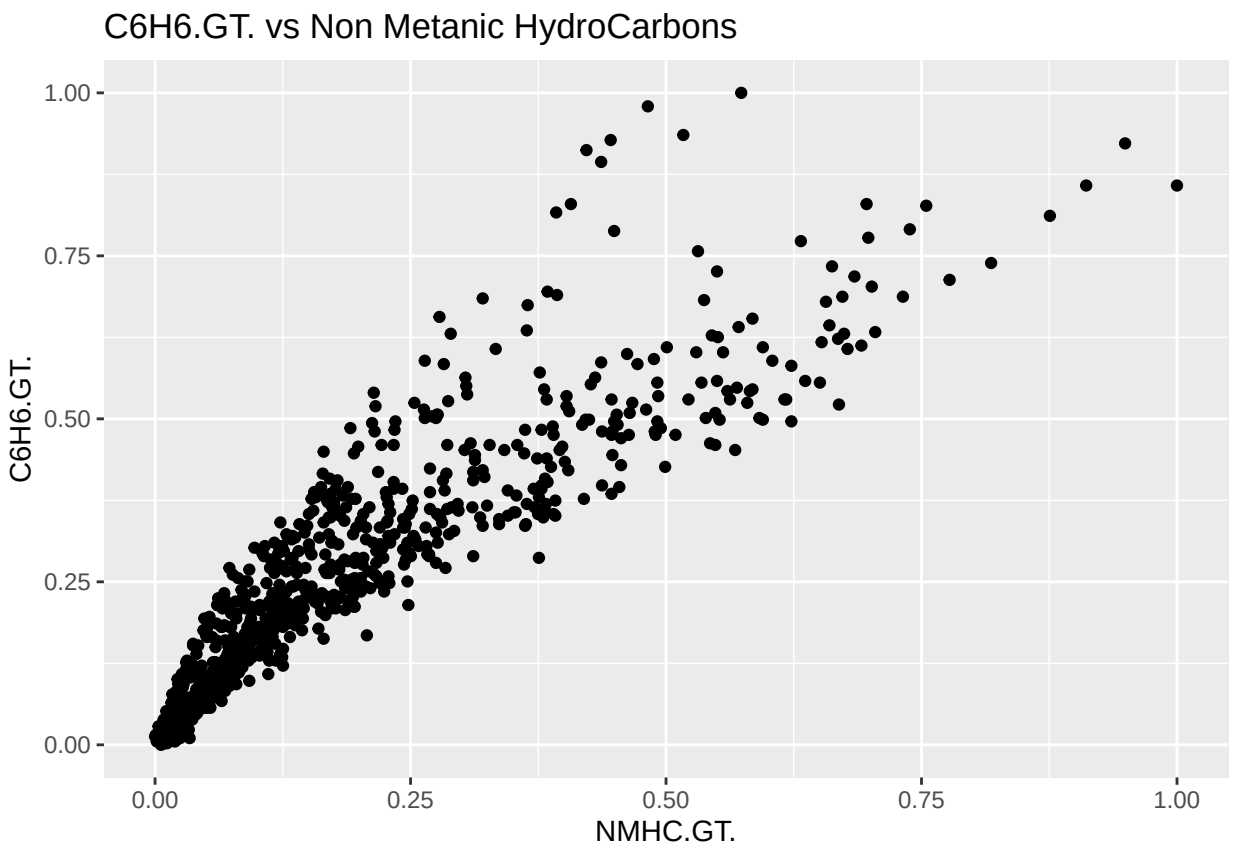
.

```
summary(CO)
```

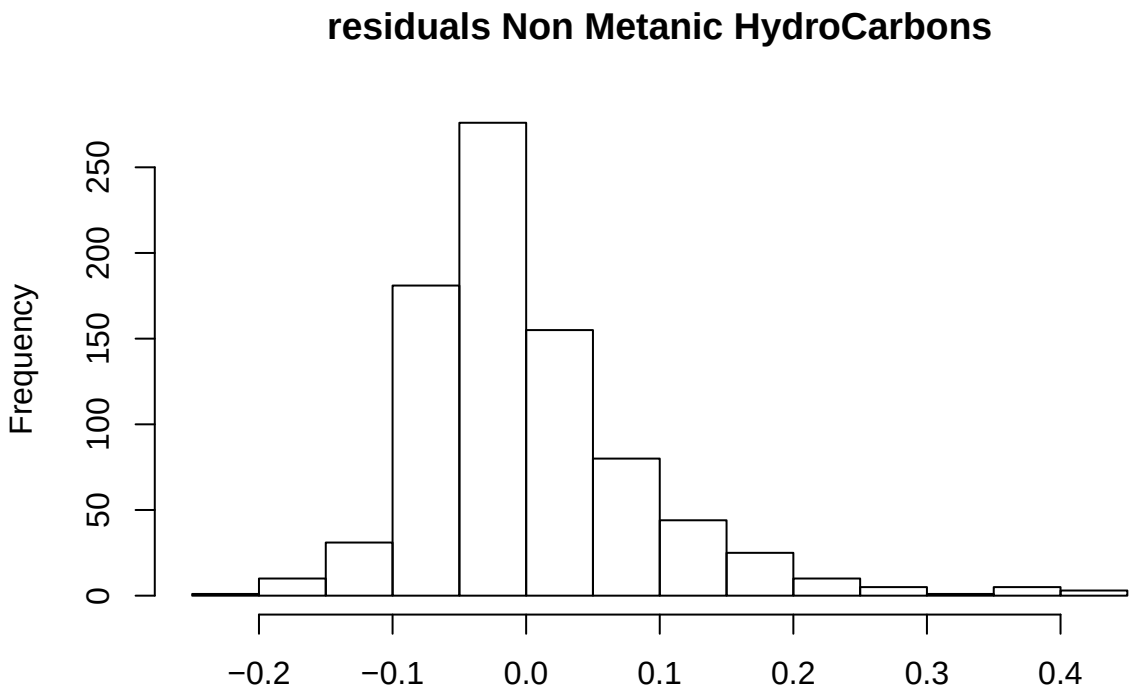
```
##
## Call:
## lm(formula = C6H6.GT. ~ CO.GT., data = airq_norm)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.246446 -0.024653 -0.002748  0.021429  0.175606
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006234   0.002738  -2.277   0.0231 *
## CO.GT.       1.031752   0.008577 120.298   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04454 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
```

#Non Metanic HydroCarbons (NMHC.GT.)

```
ggplot(airq_norm, aes(x = NMHC.GT.,
                      y = C6H6.GT.)) +
  geom_point() +
  ggtitle("C6H6.GT. vs Non Metanic HydroCarbons")
```



```
NMHC <- lm(C6H6.GT. ~ NMHC.GT., airq_norm)
residuals(NMHC) %>% hist(main = "residuals Non Metanic HydroCarbons")
```



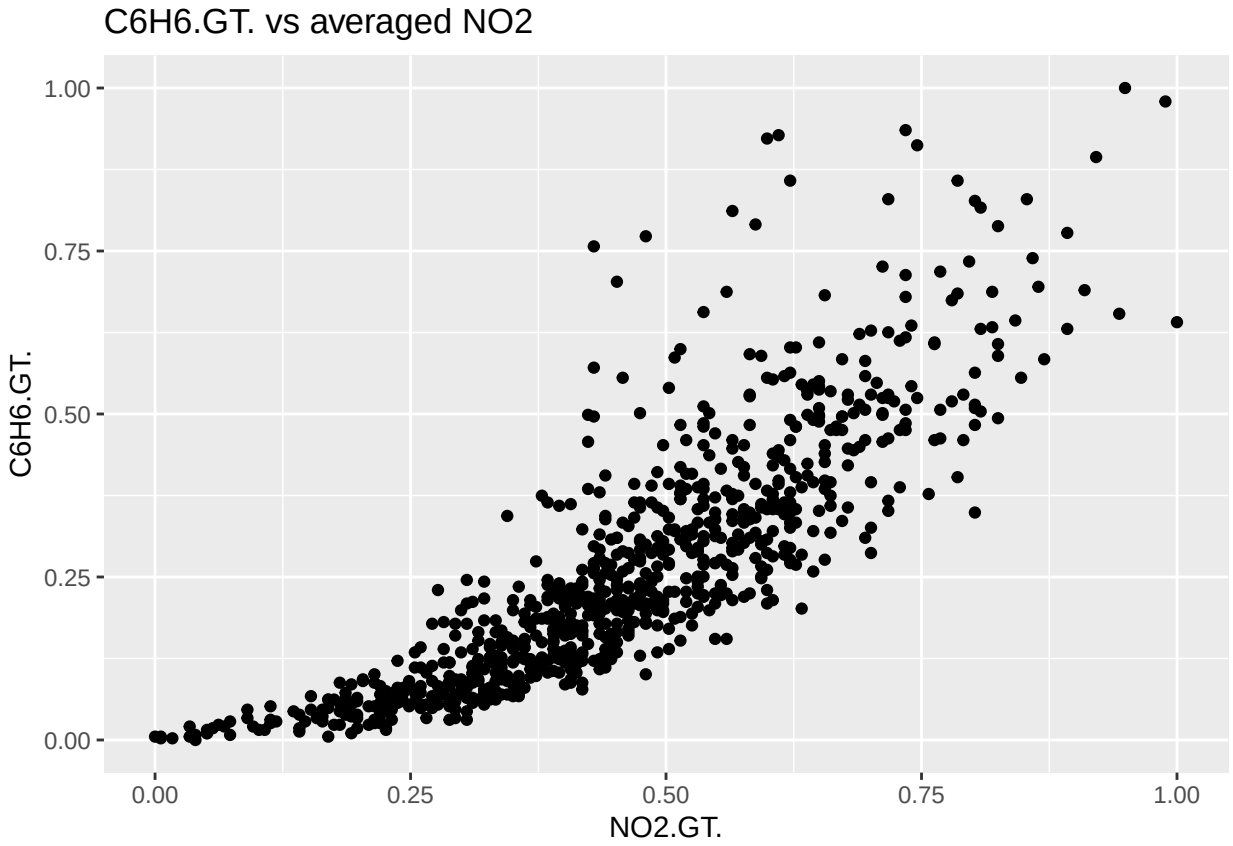
.

```
summary(NMHC)
```

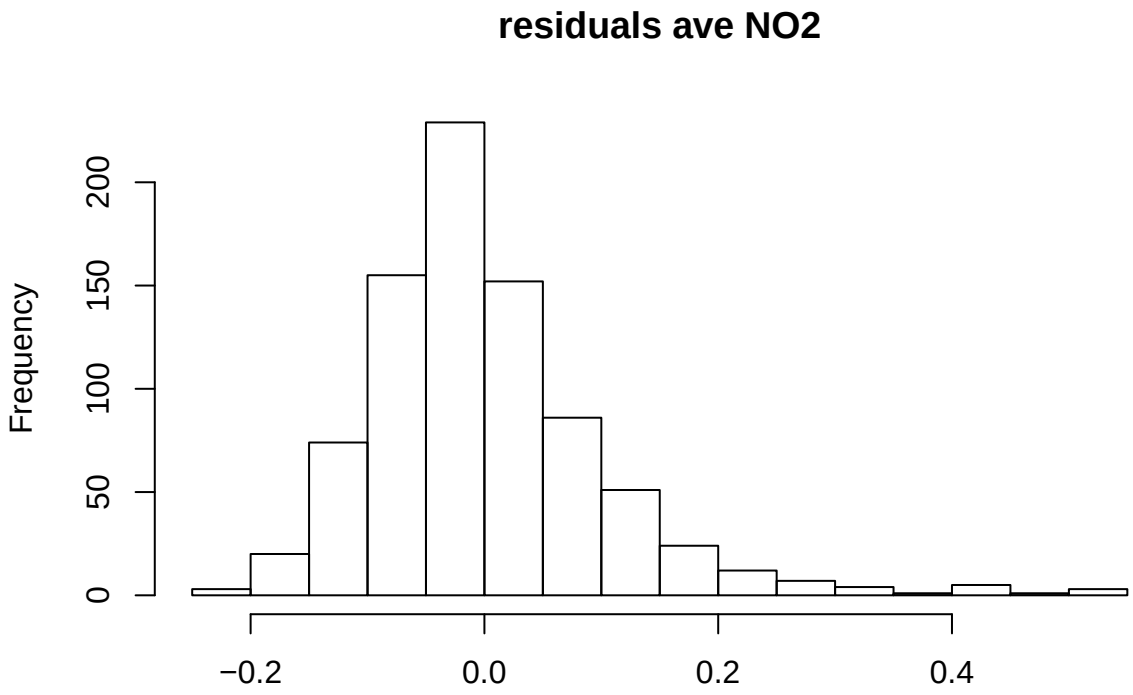
```
##
## Call:
## lm(formula = C6H6.GT. ~ NMHC.GT., data = airq_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21156 -0.05312 -0.01712  0.03570  0.42827
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.080435   0.004311   18.66   <2e-16 ***
## NMHC.GT.    0.975924   0.016655   58.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08442 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
```

#ave NO2 (NO2.GT.)

```
ggplot(airq_norm, aes(x = NO2.GT.,
                      y = C6H6.GT.)) +
  geom_point() +
  ggtitle("C6H6.GT. vs averaged NO2")
```




```
NO2 <- lm(C6H6.GT. ~ NO2.GT., airq_norm)
residuals(NO2) %>% hist(main = "residuals ave NO2")
```



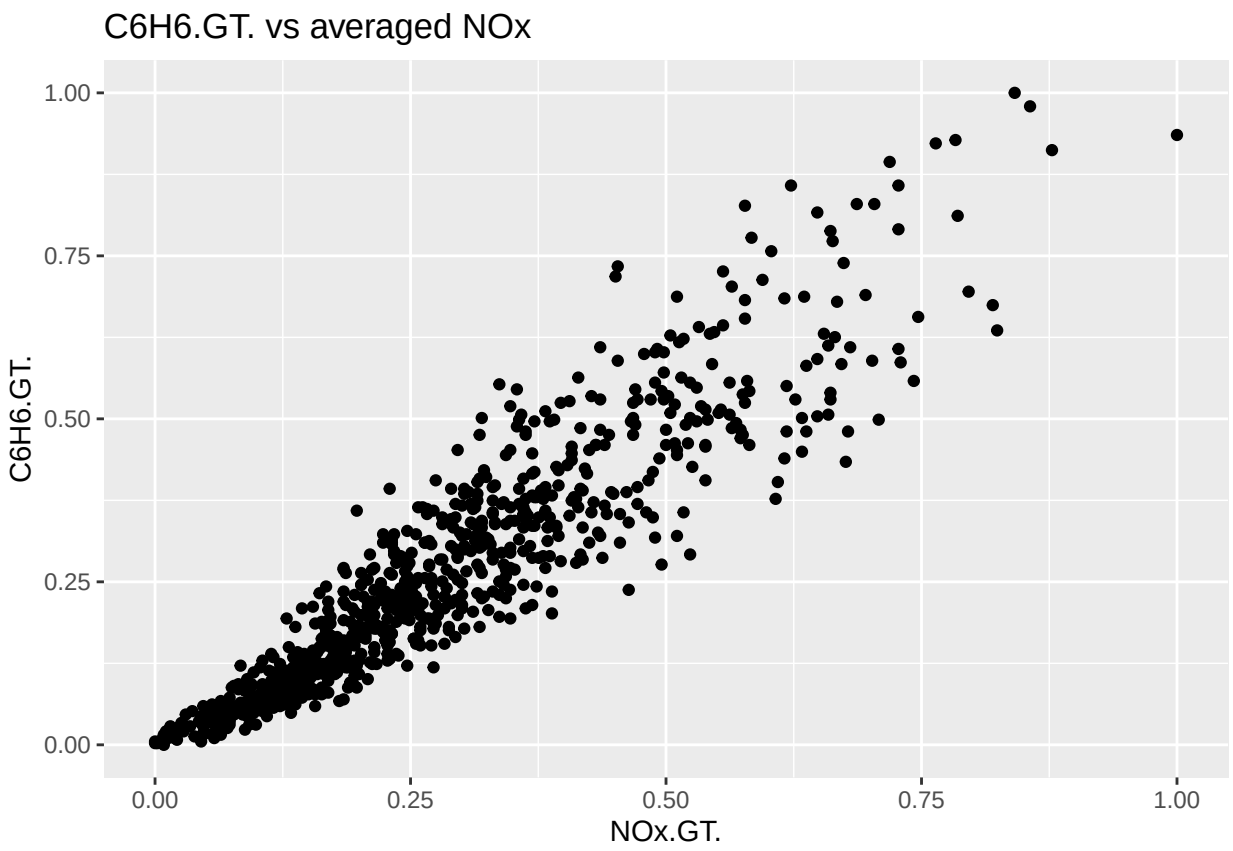
.

```
summary(NO2)
```

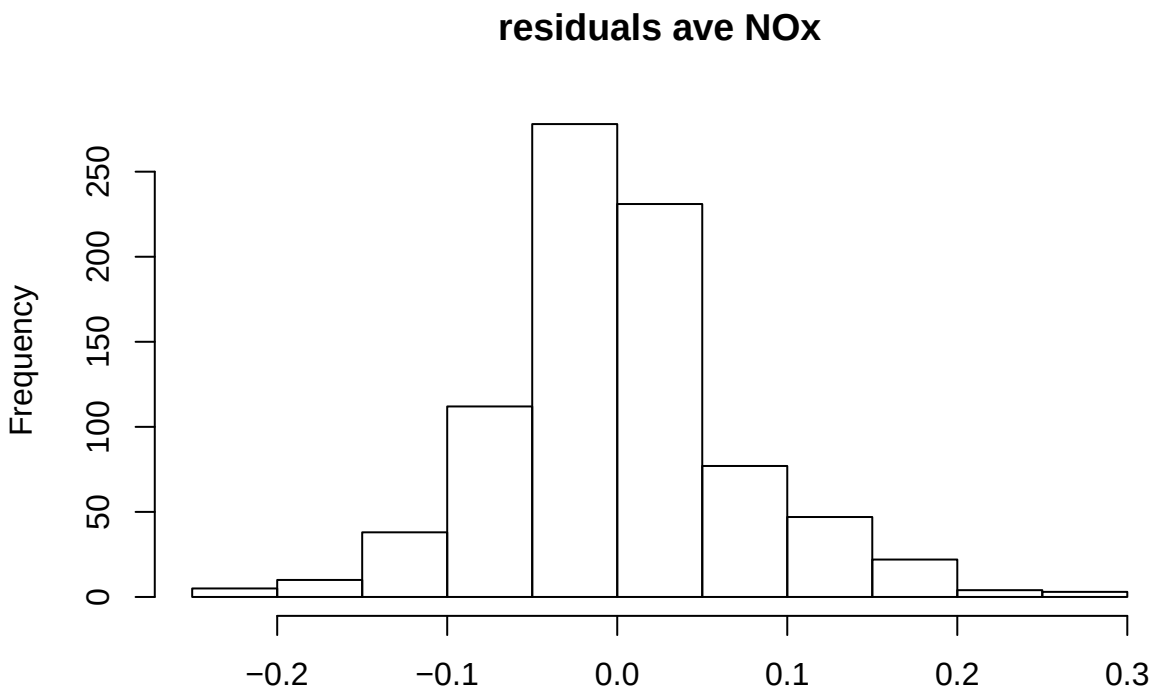
```
##
## Call:
## lm(formula = C6H6.GT. ~ NO2.GT., data = airq_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22960 -0.06523 -0.01512  0.04535  0.52958
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.153377   0.009824  -15.61   <2e-16 ***
## NO2.GT.      0.912185   0.019953   45.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.102 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
```

#ave NOx (NO2.GT.)

```
ggplot(airq_norm, aes(x = NOx.GT.,
                      y = C6H6.GT.)) +
  geom_point() +
  ggtitle("C6H6.GT. vs averaged NOx")
```



```
NOx <- lm(C6H6.GT. ~ NOx.GT., airq_norm)
residuals(NOx) %>% hist(main = "residuals ave NOx")
```



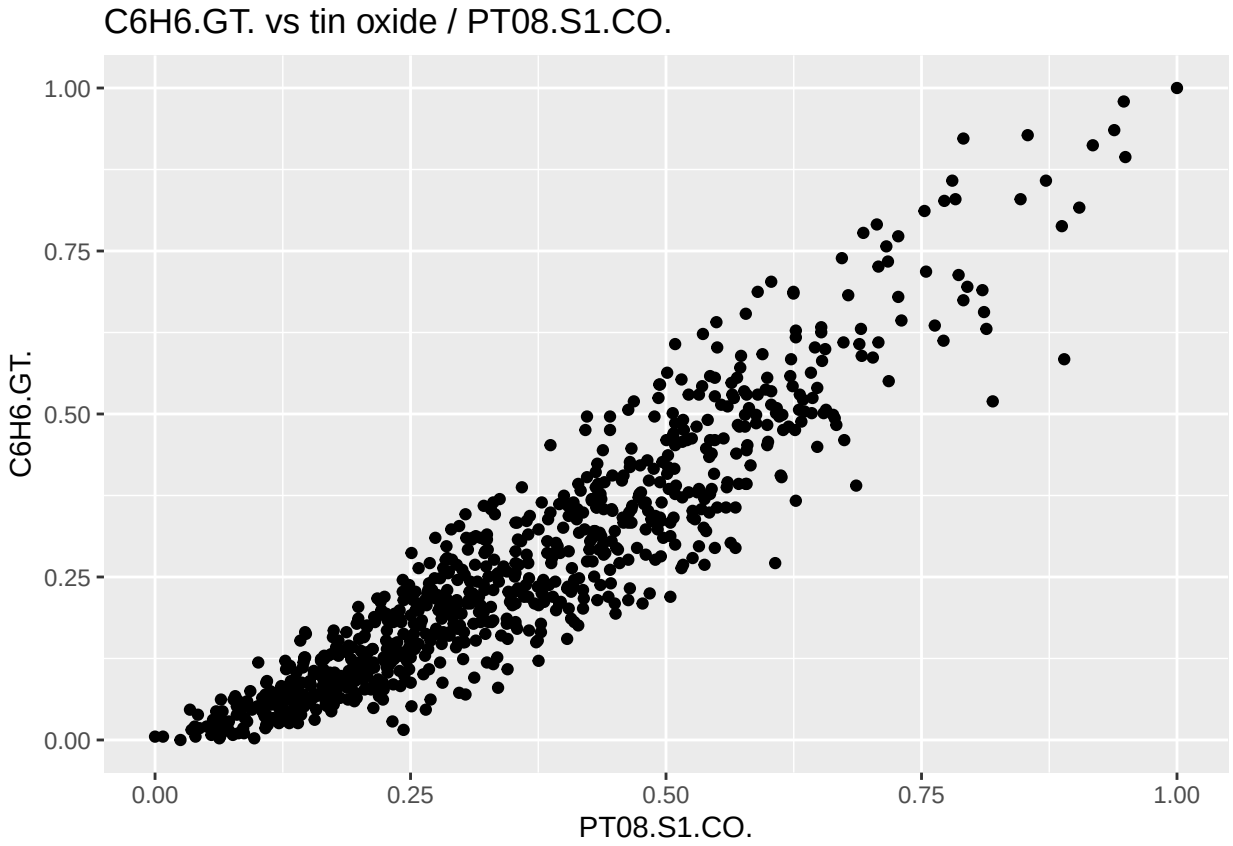
.

```
summary(NOx)
```

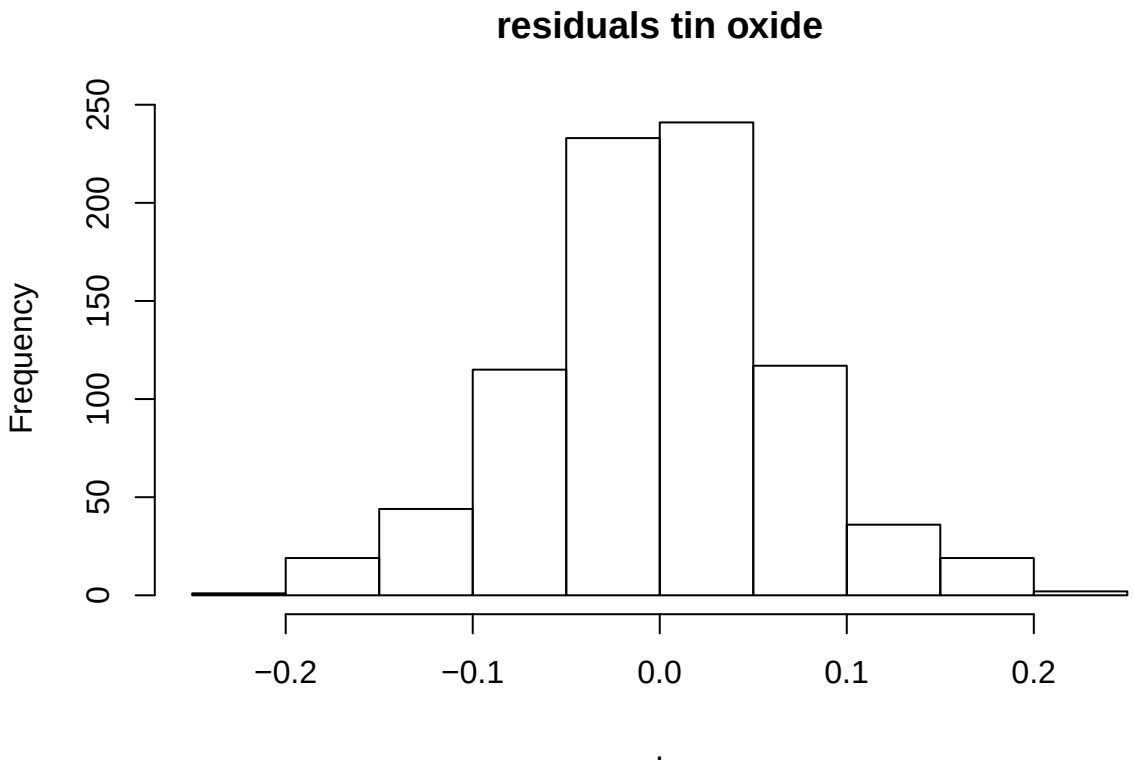
```
##
## Call:
## lm(formula = C6H6.GT. ~ NOx.GT., data = airq_norm)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.229885 -0.039333 -0.004928  0.032293  0.295763
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.020242   0.004727  -4.282 2.07e-05 ***
## NOx.GT.      1.012233   0.014225  71.157  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07179 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
```

#ave tin oxide (PT08.S1.CO.)

```
ggplot(airq_norm, aes(x = PT08.S1.CO.,
                      y = C6H6.GT.)) +
  geom_point() +
  ggtitle("C6H6.GT. vs tin oxide / PT08.S1.CO.")
```




```
PT08.S1 <- lm(C6H6.GT. ~ PT08.S1.CO., airq_norm)
residuals(PT08.S1) %>% hist(main = "residuals tin oxide")
```



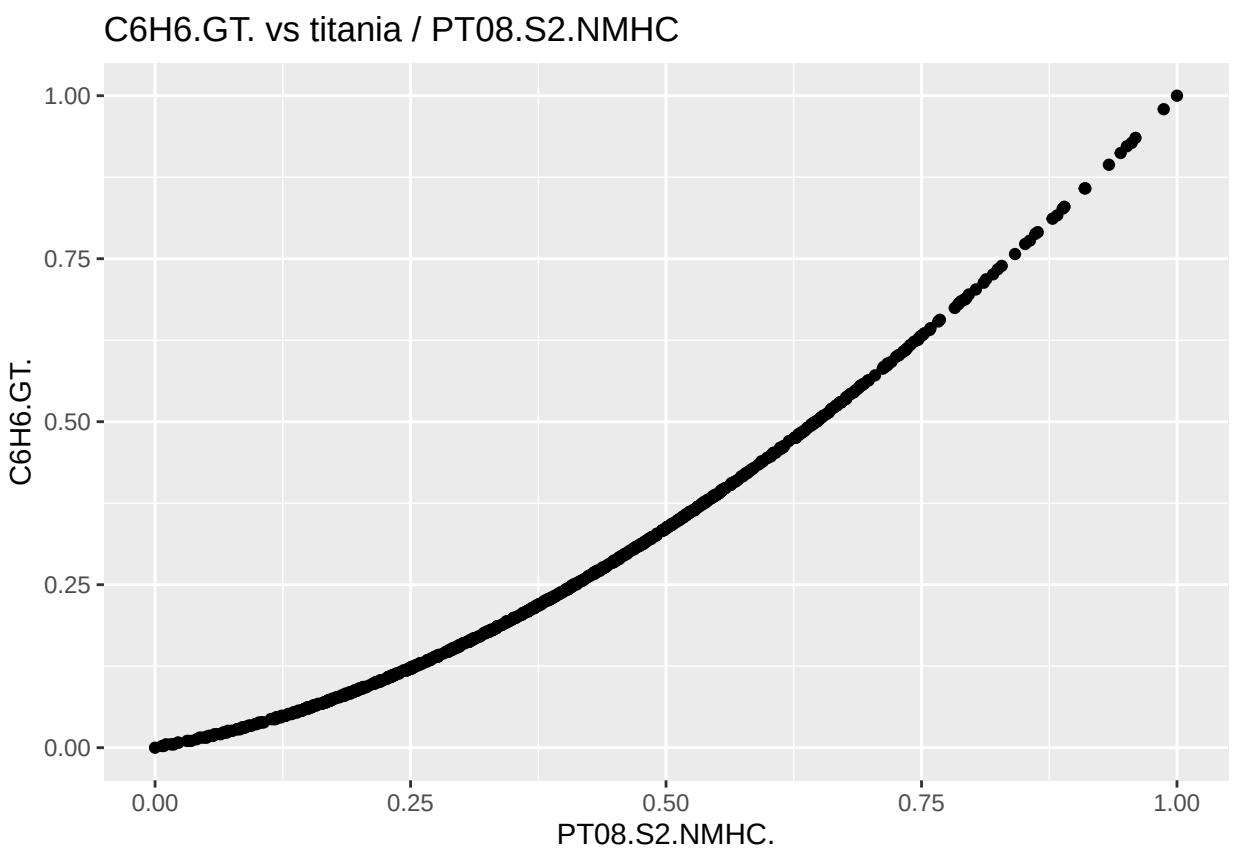
.

```
summary(PT08.S1)
```

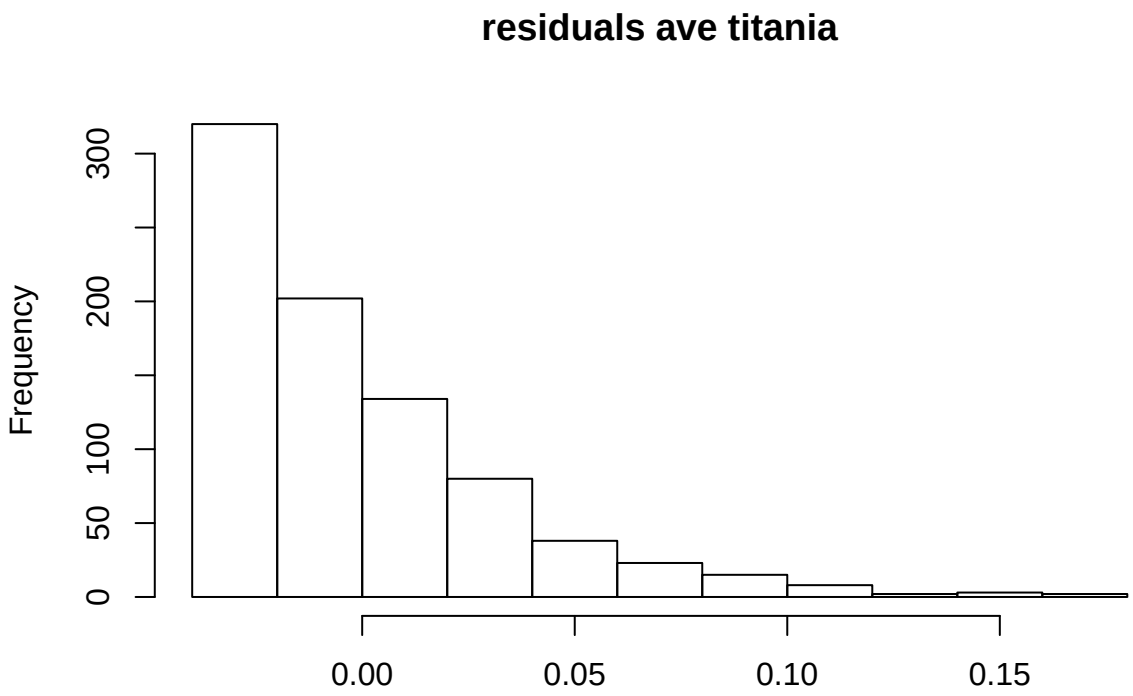
```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S1.CO., data = airq_norm)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.234852 -0.041978  0.000656  0.042554  0.241339
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.070423   0.005175  -13.61   <2e-16 ***
## PT08.S1.CO.  0.950160   0.012931   73.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06983 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
```

#ave titania (PT08.S2.NMHC)

```
ggplot(airq_norm, aes(x = PT08.S2.NMHC.,
                      y = C6H6.GT.)) +
  geom_point() +
  ggtitle("C6H6.GT. vs titania / PT08.S2.NMHC")
```



```
PT08.S2 <- lm(C6H6.GT. ~ PT08.S2.NMHC., airq_norm)
residuals(PT08.S2) %>% hist(main = "residuals ave titania")
```



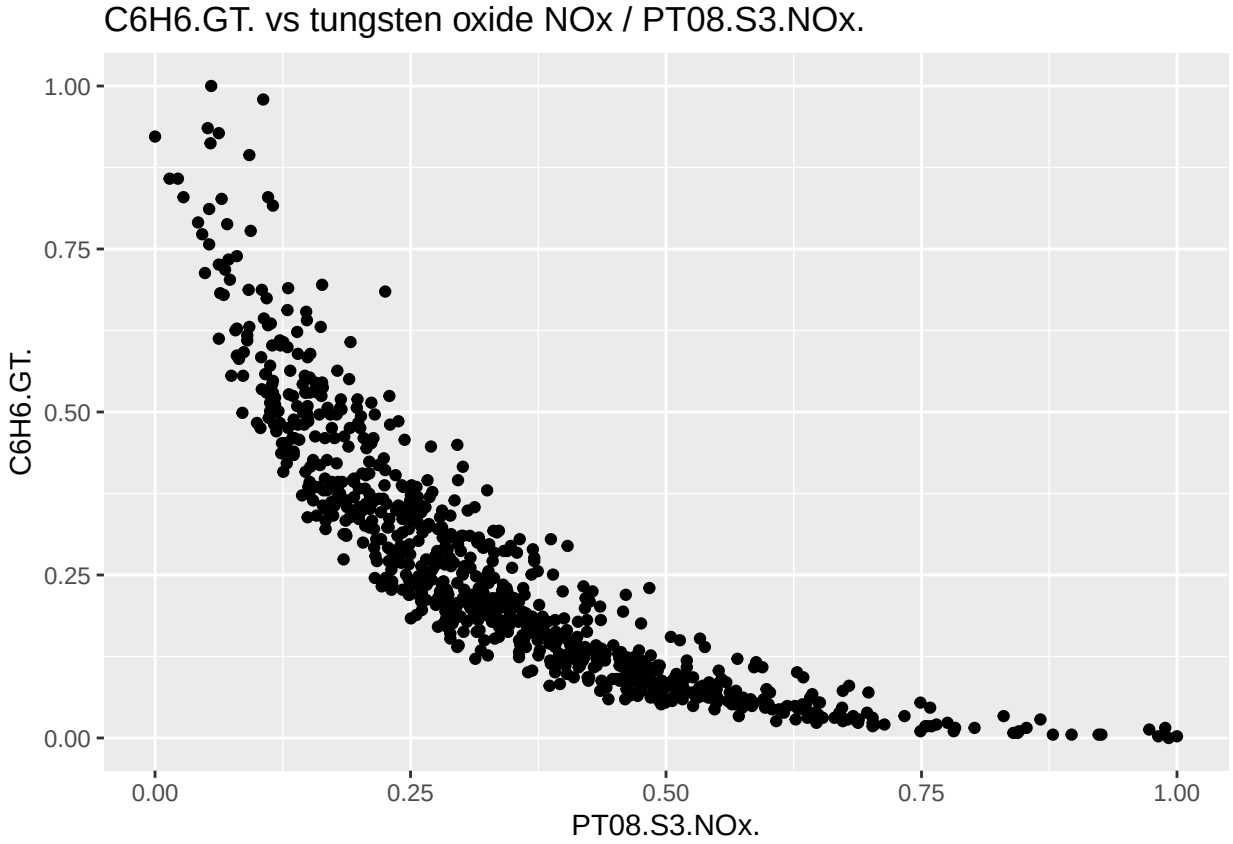
.

```
summary(PT08.S2)
```

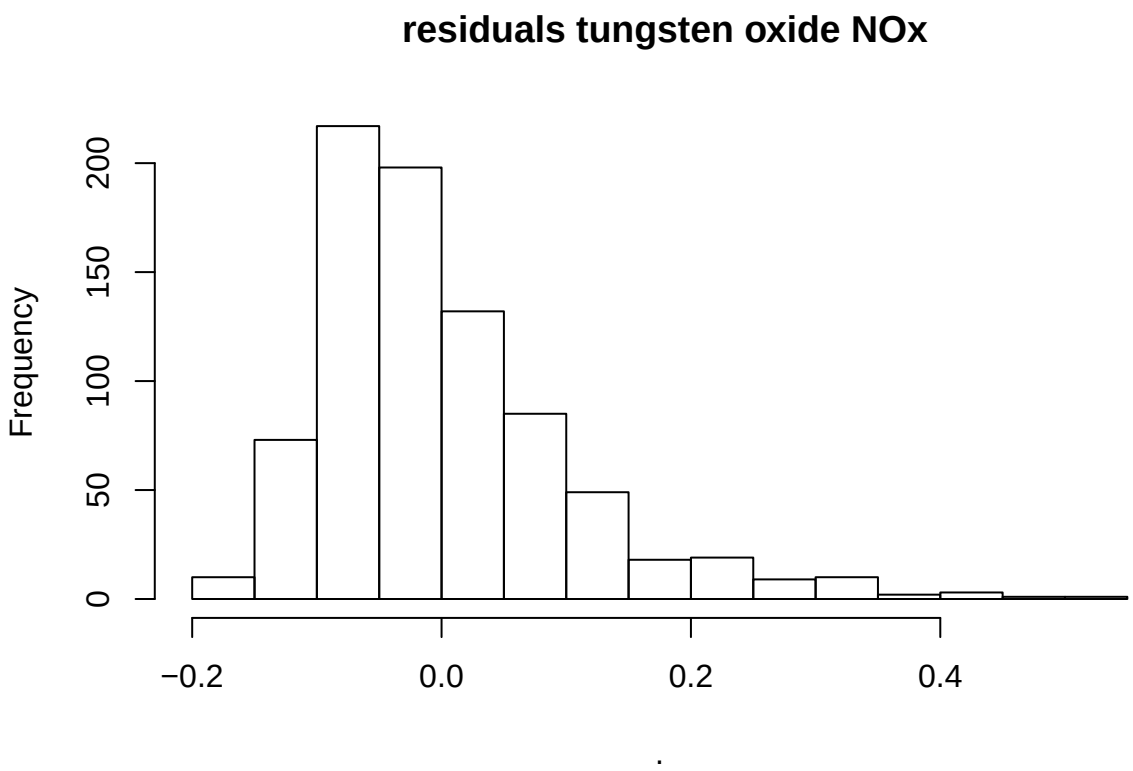
```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S2.NMHC., data = airq_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.02964 -0.02476 -0.01192  0.01419  0.17634
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.101710   0.002532  -40.18   <2e-16 ***
## PT08.S2.NMHC.  0.925371   0.005676  163.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03328 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
```

#ave tungsten oxide NOx targeted (PT08.S3.NOx.) - data not linear

```
ggplot(airq_norm, aes(x = PT08.S3.NOx.,
                      y = C6H6.GT.)) +
  geom_point() +
  ggtitle("C6H6.GT. vs tungsten oxide NOx / PT08.S3.NOx.")
```




```
PT08.S3 <- lm(C6H6.GT. ~ PT08.S3.NOx., airq_norm)
residuals(PT08.S3) %>% hist(main = " residuals tungsten oxide NOx")
```



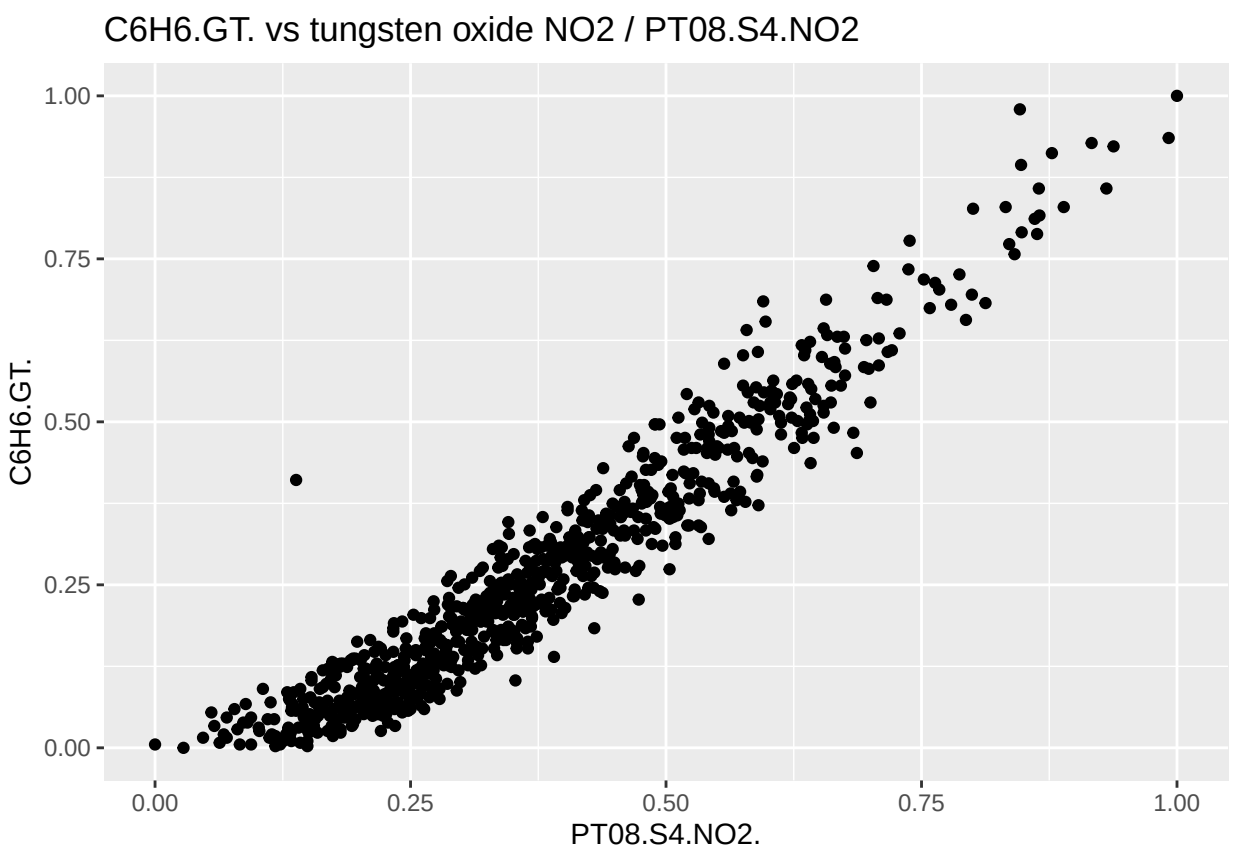
.

```
summary(PT08.S3)
```

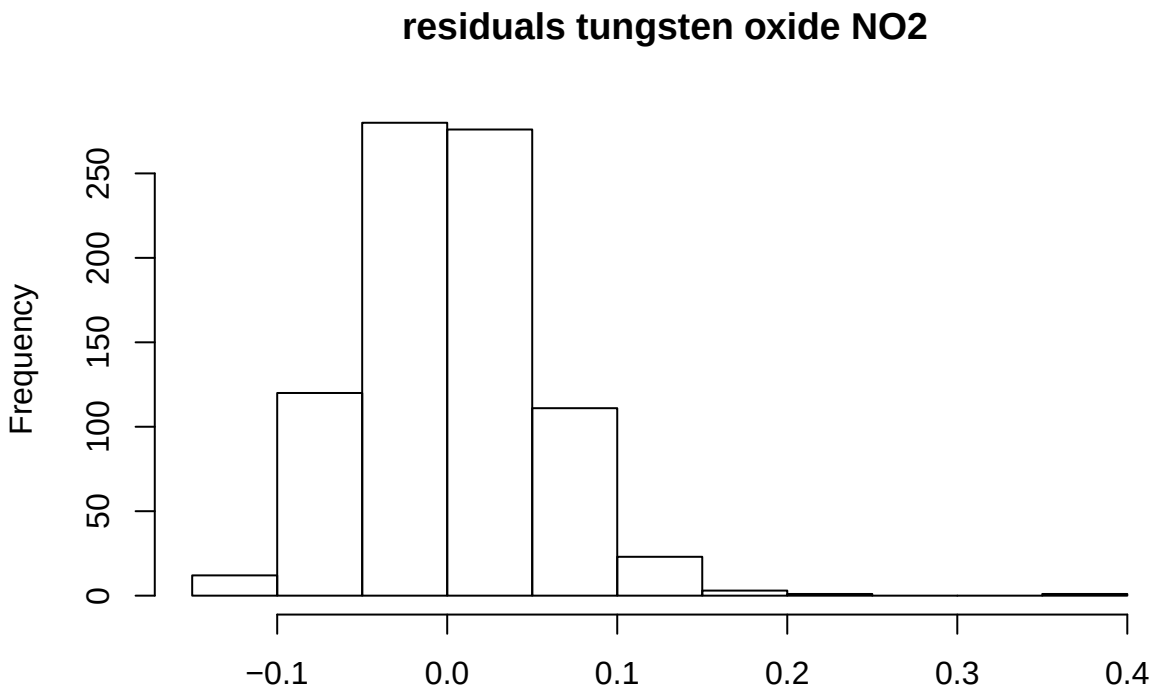
```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S3.NOx., data = airq_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16861 -0.06947 -0.02396  0.04462  0.50203
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.572754   0.007539   75.97   <2e-16 ***
## PT08.S3.NOx. -0.901926   0.019555  -46.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1014 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
```

#ave tungsten oxide NO2 targeted (PT08.S4.NO2.)

```
ggplot(airq_norm, aes(x = PT08.S4.NO2.,
                      y = C6H6.GT.)) +
  geom_point() +
  ggtitle("C6H6.GT. vs tungsten oxide NO2 / PT08.S4.NO2")
```



```
PT08.S4 <- lm(C6H6.GT. ~ PT08.S4.NO2., airq_norm)
residuals(PT08.S4) %>% hist(main = " residuals tungsten oxide NO2")
```



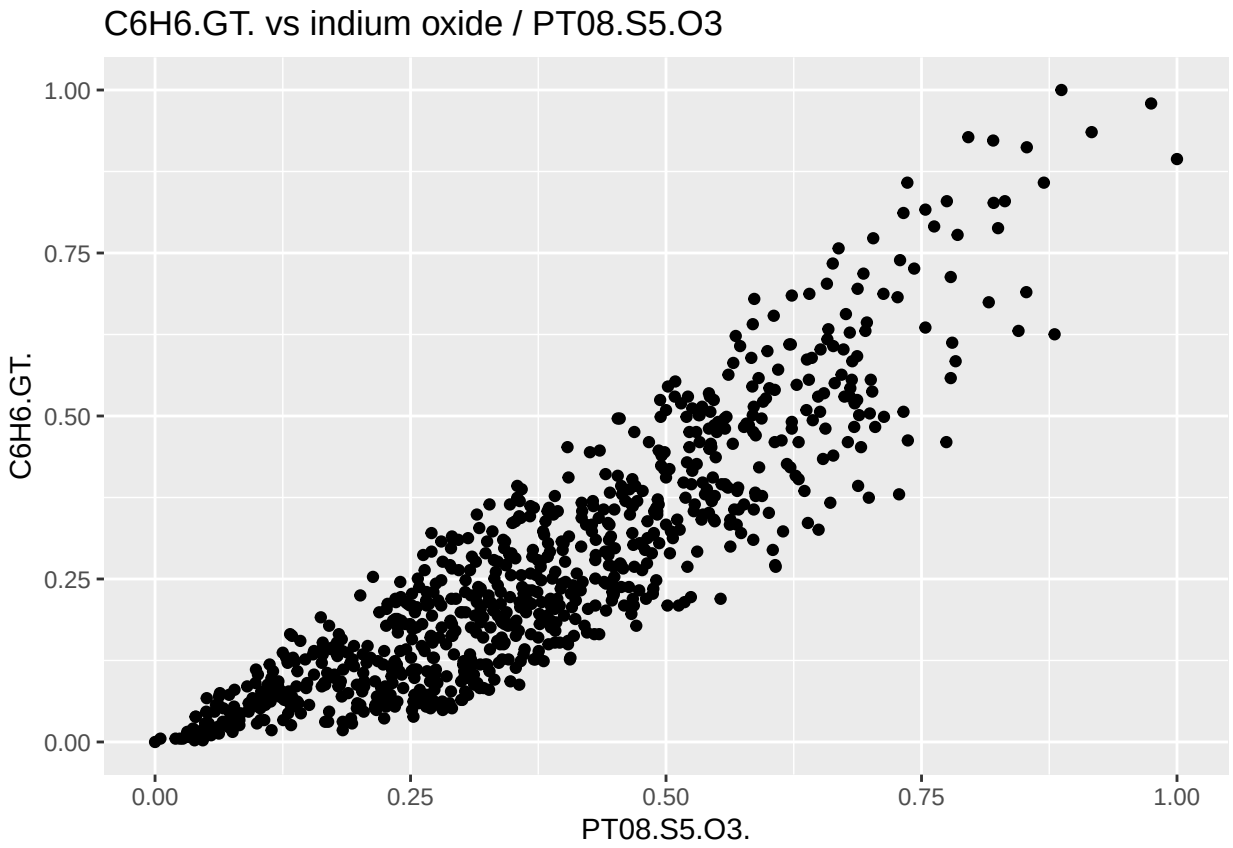
.

```
summary(PT08.S4)
```

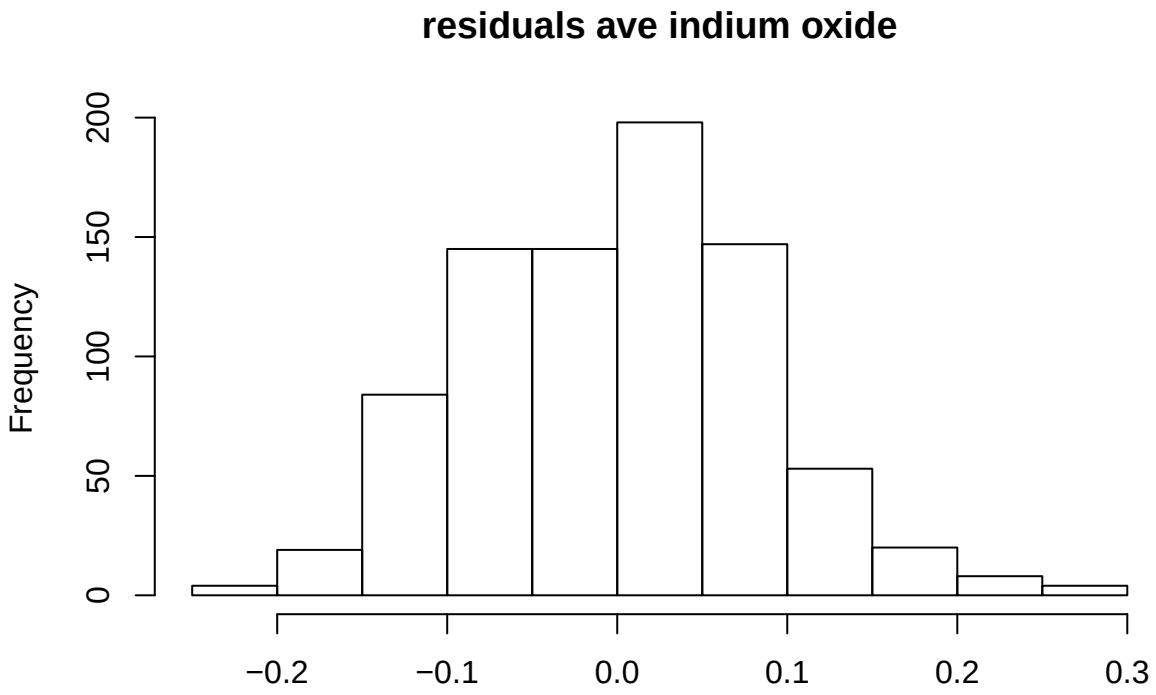
```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S4.NO2., data = airq_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14255 -0.03663  0.00027  0.03079  0.39379
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.127941   0.004362  -29.33   <2e-16 ***
## PT08.S4.NO2.  1.050347   0.010550   99.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05317 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
```

#ave indium oxide (PT08.S5.O3.)

```
ggplot(airq_norm, aes(x = PT08.S5.O3.,
                      y = C6H6.GT.)) +
  geom_point() +
  ggtitle("C6H6.GT. vs indium oxide / PT08.S5.O3")
```




```
PT08.S5 <- lm(C6H6.GT. ~ PT08.S5.O3., airq_norm)
residuals(PT08.S5) %>% hist(main = "residuals ave indium oxide")
```



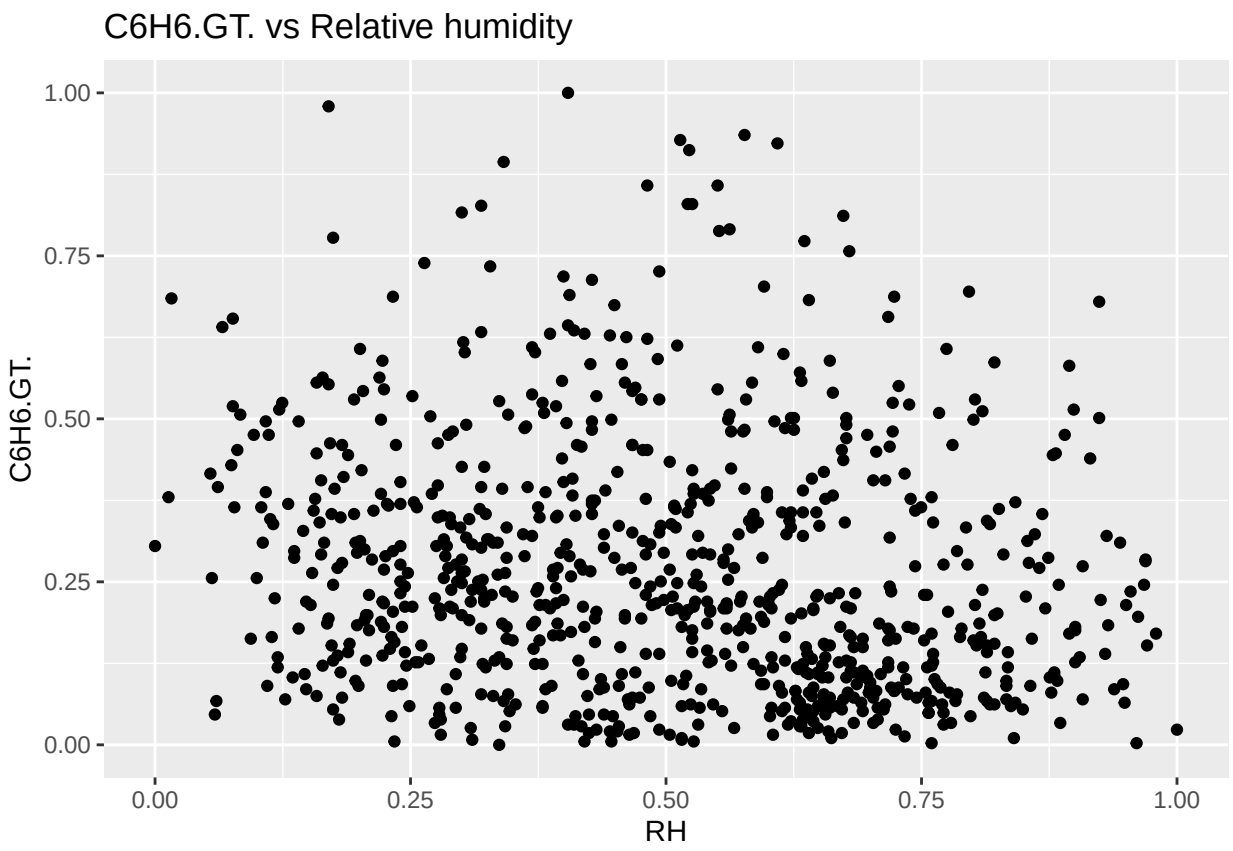
.

```
summary(PT08.S5)
```

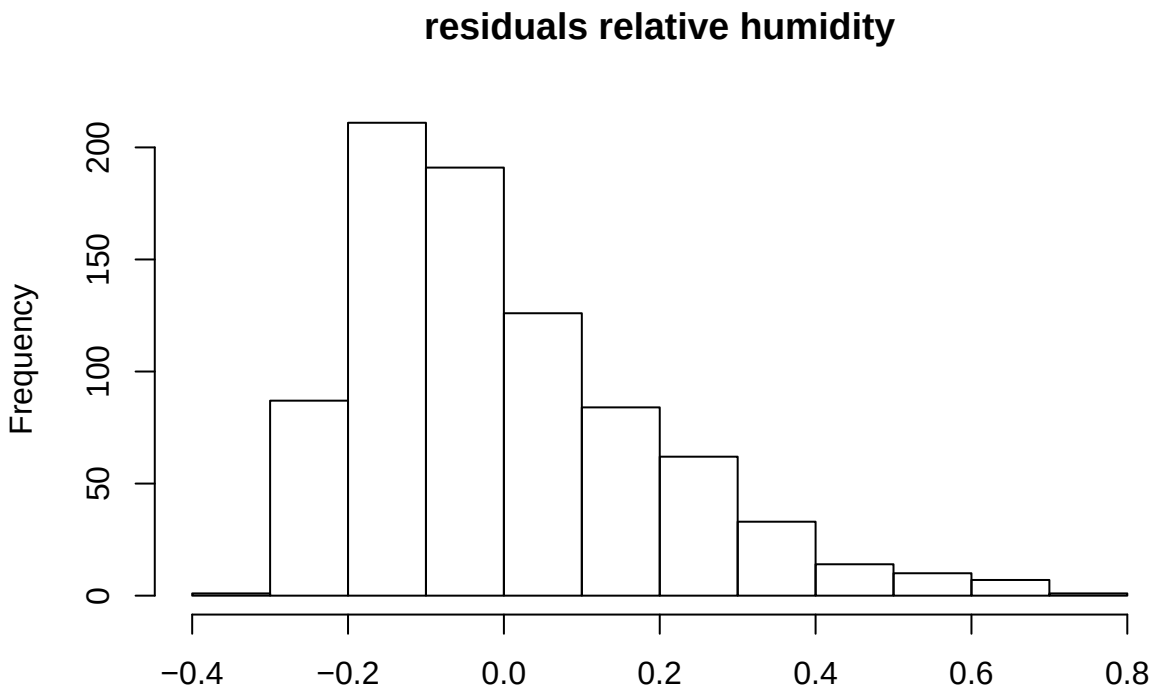
```
##
## Call:
## lm(formula = C6H6.GT. ~ PT08.S5.O3., data = airq_norm)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.207840 -0.065510  0.006316  0.056262  0.281886
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.070967   0.006481  -10.95   <2e-16 ***
## PT08.S5.O3.  0.900639   0.015454   58.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08479 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
```

#Relative humidity (RH) - data not linear

```
ggplot(airq_norm, aes(x = RH,
                      y = C6H6.GT.)) +
  geom_point() +
  ggtitle("C6H6.GT. vs Relative humidity")
```



```
RH <- lm(C6H6.GT. ~ RH, airq_norm)
residuals(RH) %>% hist(main = "residuals relative humidity")
```



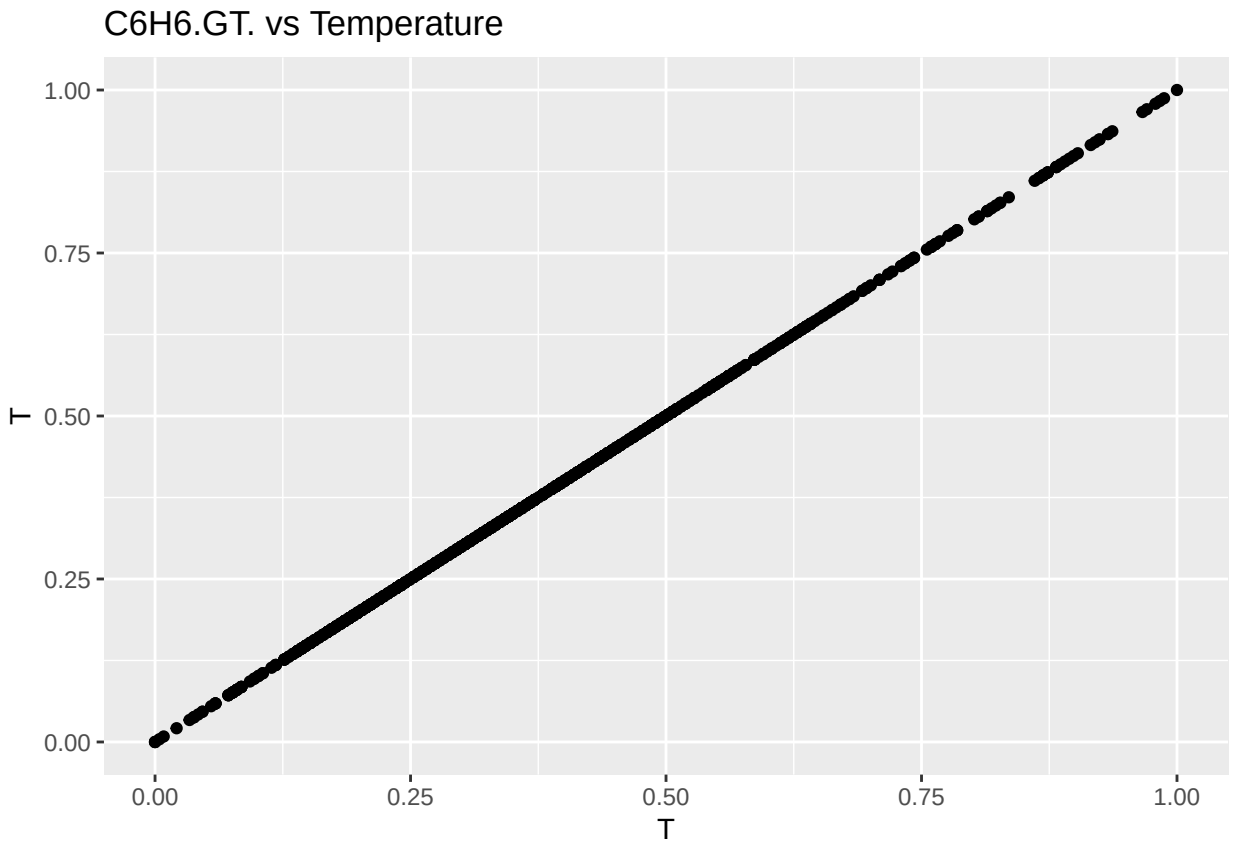
.

```
summary(RH)
```

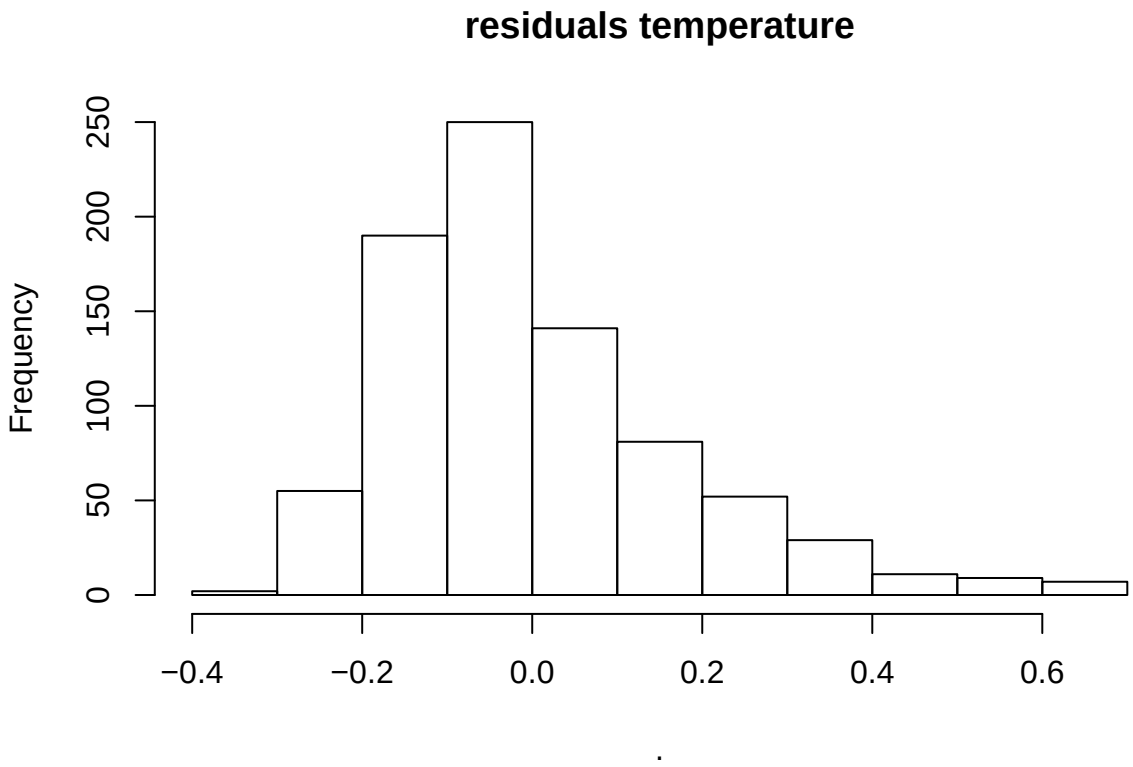
```
##
## Call:
## lm(formula = C6H6.GT. ~ RH, data = airq_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30089 -0.14383 -0.04093  0.10237  0.71992
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.34190    0.01609  21.252  < 2e-16 ***
## RH          -0.15300    0.02938  -5.208 2.41e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1887 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
```

#Temperature

```
ggplot(airq_norm, aes(x = T,
                      y = T)) +
  geom_point() +
  ggtitle("C6H6.GT. vs Temperature")
```




```
TP <- lm(C6H6.GT. ~ T, airq_norm)
residuals(TP) %>% hist(main = "residuals temperature")
```



.

```r
summary(TP)
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ T, data = airq_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32857 -0.12436 -0.03943  0.07945  0.68720
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.11080    0.01316   8.419   <2e-16 ***
## T            0.39392    0.02977  13.232   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1742 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
```

- For 2-3 of the models create train-test sets, plot the model, for the test set color real and predicted points differently; R^2 and p-value to title

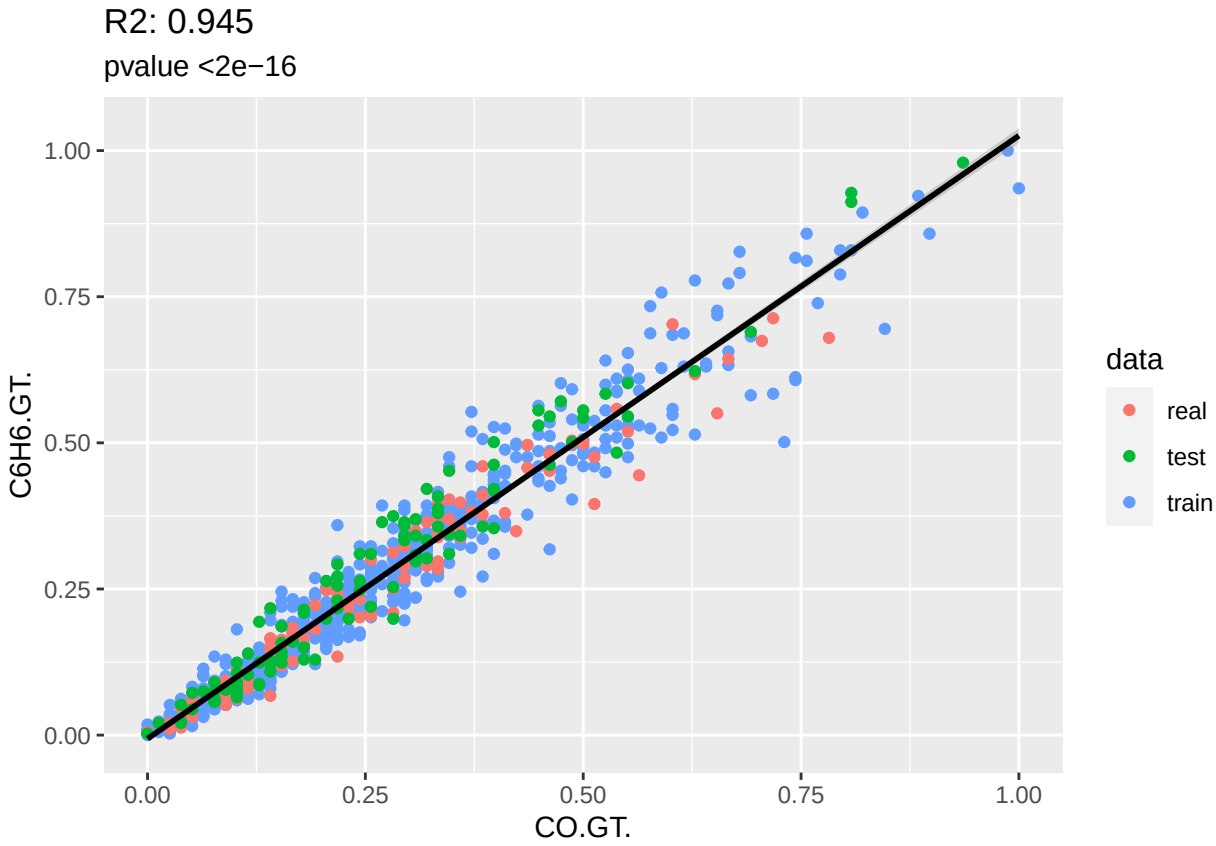
```r
#ave CO and NO2
set.seed(88)
data <- airq_norm
sample <- sample.int(n = nrow(data), size = floor(.75*nrow(data)))
train <- data[sample, ]
test <- data[-sample, ]

train_CO <- train[,c("C6H6.GT.", "CO.GT." )]
test_CO <- test[,c("C6H6.GT.", "CO.GT." )]
new_mod_CO <- lm(data = train_CO,
              C6H6.GT. ~ CO.GT.)


#summary
sum_CO <- summary(new_mod_CO)
print(sum_CO)
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ CO.GT., data = train_CO)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.244458 -0.024115 -0.002475  0.020795  0.176836
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006680   0.003172  -2.106   0.0356 *
## CO.GT.       1.029640   0.010009 102.875   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.04516 on 618 degrees of freedom
## Multiple R-squared:  0.9448, Adjusted R-squared:  0.9447
## F-statistic: 1.058e+04 on 1 and 618 DF,  p-value: < 2.2e-16
```

```r
#use model, predicts new model from newdata (test)
#predictions for new test data set, add new column
pred_CO <- predict(new_mod_CO, newdata = test_CO)
test$CO.GT._pred <- pred_CO


#make combined dataset
train_CO$data <- "train"
test_CO$data <- "test"
test_CO[1:(nrow(test_CO)/2),3] <- "real"
comb_CO <- rbind(train_CO, test_CO)

#trained and test should be similar shape
ggplot(data = comb_CO,
       aes(x = CO.GT.,
           y = C6H6.GT.,
           color = data)) +
  geom_point() +
  geom_smooth(method = "lm",
              color = "black") +
  ggtitle(paste("R2", round(sum_CO$r.squared, 3),
                sep = ": "),
          paste("pvalue <2e-16" ))
```

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




```
#same for NO2
train_NO2 <- train[,c("C6H6.GT.", "NO2.GT." )]
test_NO2 <- test[,c("C6H6.GT.", "NO2.GT." )]
mod_NO2 <- lm(data = train_NO2,
              C6H6.GT. ~ NO2.GT.)

sum_NO2 <- summary(mod_NO2)
print(sum_NO2)
```

```
##
## Call:
## lm(formula = C6H6.GT. ~ NO2.GT., data = train_NO2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19968 -0.06541 -0.01496  0.04565  0.53161
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.15667    0.01117  -14.03   <2e-16 ***
## NO2.GT.      0.91429    0.02277   40.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1012 on 618 degrees of freedom
## Multiple R-squared:  0.7229, Adjusted R-squared:  0.7225
## F-statistic:  1612 on 1 and 618 DF,  p-value: < 2.2e-16
```

```
pred_NO2 <- predict(mod_NO2, newdata = test_NO2)
test$NO2.GT._pred <- pred_NO2

train_NO2$data <- "train"
test_NO2$data <- "test"
test_NO2[1:(nrow(test_NO2)/2),3] <- "real"
comb_NO2 <- rbind(train_NO2, test_NO2)

ggplot(data = comb_NO2,
       aes(x = NO2.GT.,
           y = C6H6.GT.,
           color = data)) +
  geom_point() +
  geom_smooth(method = "lm",
              color = "black") +
  ggtitle(paste("R2", round(sum_CO$r.squared, 3),
                sep = ": "),
          paste("pvalue <2e-16" ))
```

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