Task 21

Lisa Skalon

4/9/2020

library('ggplot2')  
library('ggpubr')  
library('dplyr')  
library('tidyr')  
library('broom')  
library('lubridate')  
library('reshape2')

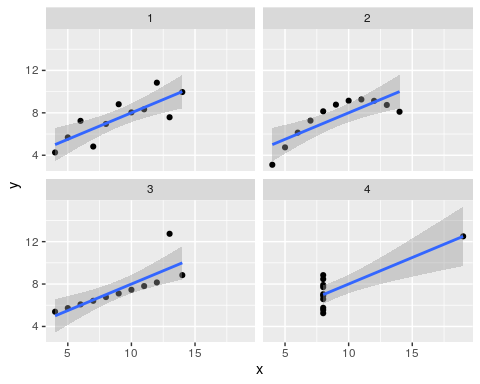
## Statistics in R: task1

### Ancomb dataset

1. Scatter plot facetted by set + the 95% confidence level interval for predictions from a linear model

# make data long  
  
df\_long <- anscombe %>%  
 mutate(index = 1:nrow(anscombe)) %>%  
 gather(key, value, -index) %>%  
 separate(key, c("var", "set"), 1, convert = TRUE) %>%  
 spread(var, value)%>%  
 select(-index)

# plot with lm line  
ggplot(df\_long,aes(x,y,group=set))+  
 geom\_point()+  
 stat\_smooth(method="lm")+  
 facet\_wrap(~set)



1. Summary by set

by(df\_long[,2:3], df\_long$set, summary)

## df\_long$set: 1  
## x y   
## Min. : 4.0 Min. : 4.260   
## 1st Qu.: 6.5 1st Qu.: 6.315   
## Median : 9.0 Median : 7.580   
## Mean : 9.0 Mean : 7.501   
## 3rd Qu.:11.5 3rd Qu.: 8.570   
## Max. :14.0 Max. :10.840   
## ------------------------------------------------------------   
## df\_long$set: 2  
## x y   
## Min. : 4.0 Min. :3.100   
## 1st Qu.: 6.5 1st Qu.:6.695   
## Median : 9.0 Median :8.140   
## Mean : 9.0 Mean :7.501   
## 3rd Qu.:11.5 3rd Qu.:8.950   
## Max. :14.0 Max. :9.260   
## ------------------------------------------------------------   
## df\_long$set: 3  
## x y   
## Min. : 4.0 Min. : 5.39   
## 1st Qu.: 6.5 1st Qu.: 6.25   
## Median : 9.0 Median : 7.11   
## Mean : 9.0 Mean : 7.50   
## 3rd Qu.:11.5 3rd Qu.: 7.98   
## Max. :14.0 Max. :12.74   
## ------------------------------------------------------------   
## df\_long$set: 4  
## x y   
## Min. : 8 Min. : 5.250   
## 1st Qu.: 8 1st Qu.: 6.170   
## Median : 8 Median : 7.040   
## Mean : 9 Mean : 7.501   
## 3rd Qu.: 8 3rd Qu.: 8.190   
## Max. :19 Max. :12.500

1. Correlation betveen x and y in each set. Parametric and non-parametric tests.

pearson <- df\_long %>%   
 group\_by(set) %>%   
 do(tidy(cor.test(.$x, .$y, method='pearson')))  
  
spearman <- df\_long %>%   
 group\_by(set) %>%   
 do(tidy(cor.test(.$x, .$y, method='spearman')))  
  
df\_cor <- data.frame(pearson$estimate, pearson$p.value, spearman$estimate, spearman$p.value)  
df\_cor

## pearson.estimate pearson.p.value spearman.estimate spearman.p.value  
## 1 0.8164205 0.002169629 0.8181818 0.003734471  
## 2 0.8162365 0.002178816 0.6909091 0.023058874  
## 3 0.8162867 0.002176305 0.9909091 0.000000000  
## 4 0.8165214 0.002164602 0.5000000 0.117306803

### Air quality dataset

1. Clean dataset

# read  
df <- read.csv("./AirQualityUCI.csv", stringsAsFactors = FALSE, sep = ';')  
  
# examine the data structure  
str(df)

## 'data.frame': 9471 obs. of 17 variables:  
## $ Date : chr "10/03/2004" "10/03/2004" "10/03/2004" "10/03/2004" ...  
## $ Time : chr "18.00.00" "19.00.00" "20.00.00" "21.00.00" ...  
## $ CO.GT. : chr "2,6" "2" "2,2" "2,2" ...  
## $ 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. : chr "11,9" "9,4" "9,0" "9,2" ...  
## $ 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 : chr "13,6" "13,3" "11,9" "11,0" ...  
## $ RH : chr "48,9" "47,7" "54,0" "60,0" ...  
## $ AH : chr "0,7578" "0,7255" "0,7502" "0,7867" ...  
## $ X : logi NA NA NA NA NA NA ...  
## $ X.1 : logi NA NA NA NA NA NA ...

# convert date and time to special type  
df$date\_time = dmy\_hms(paste(df$Date, df$Time))  
  
# remove unuseful cols  
df <- df[,c(3:15,18) ]  
  
# chr to numeric  
char\_columns <- sapply(df, is.character)  
df[ , char\_columns] <- lapply(df[ , char\_columns] , function(x) as.numeric(gsub(",", ".", x)))  
  
# explore summary   
summary(df)

## CO.GT. PT08.S1.CO. NMHC.GT. C6H6.GT.   
## Min. :-200.00 Min. :-200 Min. :-200.0 Min. :-200.000   
## 1st Qu.: 0.60 1st Qu.: 921 1st Qu.:-200.0 1st Qu.: 4.000   
## Median : 1.50 Median :1053 Median :-200.0 Median : 7.900   
## Mean : -34.21 Mean :1049 Mean :-159.1 Mean : 1.866   
## 3rd Qu.: 2.60 3rd Qu.:1221 3rd Qu.:-200.0 3rd Qu.: 13.600   
## Max. : 11.90 Max. :2040 Max. :1189.0 Max. : 63.700   
## NA's :114 NA's :114 NA's :114 NA's :114   
## PT08.S2.NMHC. NOx.GT. PT08.S3.NOx. NO2.GT.   
## Min. :-200.0 Min. :-200.0 Min. :-200 Min. :-200.00   
## 1st Qu.: 711.0 1st Qu.: 50.0 1st Qu.: 637 1st Qu.: 53.00   
## Median : 895.0 Median : 141.0 Median : 794 Median : 96.00   
## Mean : 894.6 Mean : 168.6 Mean : 795 Mean : 58.15   
## 3rd Qu.:1105.0 3rd Qu.: 284.0 3rd Qu.: 960 3rd Qu.: 133.00   
## Max. :2214.0 Max. :1479.0 Max. :2683 Max. : 340.00   
## NA's :114 NA's :114 NA's :114 NA's :114   
## PT08.S4.NO2. PT08.S5.O3. T RH   
## Min. :-200 Min. :-200.0 Min. :-200.000 Min. :-200.00   
## 1st Qu.:1185 1st Qu.: 700.0 1st Qu.: 10.900 1st Qu.: 34.10   
## Median :1446 Median : 942.0 Median : 17.200 Median : 48.60   
## Mean :1391 Mean : 975.1 Mean : 9.778 Mean : 39.49   
## 3rd Qu.:1662 3rd Qu.:1255.0 3rd Qu.: 24.100 3rd Qu.: 61.90   
## Max. :2775 Max. :2523.0 Max. : 44.600 Max. : 88.70   
## NA's :114 NA's :114 NA's :114 NA's :114   
## AH date\_time   
## Min. :-200.0000 Min. :2004-03-10 18:00:00   
## 1st Qu.: 0.6923 1st Qu.:2004-06-16 05:00:00   
## Median : 0.9768 Median :2004-09-21 16:00:00   
## Mean : -6.8376 Mean :2004-09-21 16:00:00   
## 3rd Qu.: 1.2962 3rd Qu.:2004-12-28 03:00:00   
## Max. : 2.2310 Max. :2005-04-04 14:00:00   
## NA's :114 NA's :114

head(df)

## CO.GT. PT08.S1.CO. NMHC.GT. C6H6.GT. PT08.S2.NMHC. NOx.GT. PT08.S3.NOx.  
## 1 2.6 1360 150 11.9 1046 166 1056  
## 2 2.0 1292 112 9.4 955 103 1174  
## 3 2.2 1402 88 9.0 939 131 1140  
## 4 2.2 1376 80 9.2 948 172 1092  
## 5 1.6 1272 51 6.5 836 131 1205  
## 6 1.2 1197 38 4.7 750 89 1337  
## NO2.GT. PT08.S4.NO2. PT08.S5.O3. T RH AH date\_time  
## 1 113 1692 1268 13.6 48.9 0.7578 2004-03-10 18:00:00  
## 2 92 1559 972 13.3 47.7 0.7255 2004-03-10 19:00:00  
## 3 114 1555 1074 11.9 54.0 0.7502 2004-03-10 20:00:00  
## 4 122 1584 1203 11.0 60.0 0.7867 2004-03-10 21:00:00  
## 5 116 1490 1110 11.2 59.6 0.7888 2004-03-10 22:00:00  
## 6 96 1393 949 11.2 59.2 0.7848 2004-03-10 23:00:00

# remove na  
df <- na.omit(df)  
  
# value -200 is suspicious  
# let values -200 be NA  
df\_no200 <- df[, 1:13]  
df\_no200[] <- sapply(df[, 1:13] , function(x) {x[grep("-200", x)] = NA; return((x))})   
str(df\_no200)

## 'data.frame': 9357 obs. of 13 variables:  
## $ CO.GT. : num 2.6 2 2.2 2.2 1.6 1.2 1.2 1 0.9 0.6 ...  
## $ PT08.S1.CO. : num 1360 1292 1402 1376 1272 ...  
## $ NMHC.GT. : num 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.: num 1046 955 939 948 836 ...  
## $ NOx.GT. : num 166 103 131 172 131 89 62 62 45 NA ...  
## $ PT08.S3.NOx. : num 1056 1174 1140 1092 1205 ...  
## $ NO2.GT. : num 113 92 114 122 116 96 77 76 60 NA ...  
## $ PT08.S4.NO2. : num 1692 1559 1555 1584 1490 ...  
## $ PT08.S5.O3. : num 1268 972 1074 1203 1110 ...  
## $ 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 ...

# replace na with median   
df\_named <- replace(df\_no200, TRUE, lapply(df\_no200, function(x) replace(x, is.na(x), median(x, na.rm = TRUE))))  
  
# new summary without outliers  
summary(df\_named)

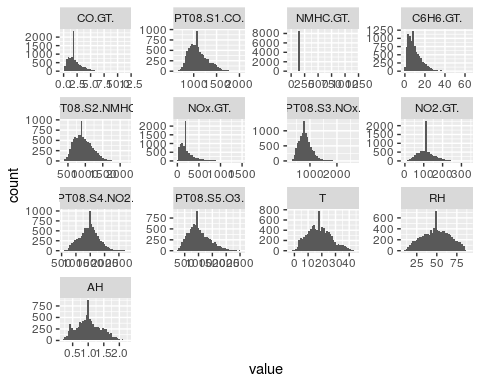
## CO.GT. PT08.S1.CO. NMHC.GT. C6H6.GT.   
## Min. : 0.100 Min. : 647 Min. : 7.0 Min. : 0.10   
## 1st Qu.: 1.200 1st Qu.: 941 1st Qu.: 150.0 1st Qu.: 4.60   
## Median : 1.800 Median :1063 Median : 150.0 Median : 8.20   
## Mean : 2.089 Mean :1098 Mean : 156.7 Mean :10.01   
## 3rd Qu.: 2.600 3rd Qu.:1221 3rd Qu.: 150.0 3rd Qu.:13.60   
## Max. :11.900 Max. :2040 Max. :1189.0 Max. :63.70   
## PT08.S2.NMHC. NOx.GT. PT08.S3.NOx. NO2.GT.   
## Min. : 383 Min. : 2.0 Min. : 322.0 Min. : 2.0   
## 1st Qu.: 743 1st Qu.: 112.0 1st Qu.: 666.0 1st Qu.: 86.0   
## Median : 909 Median : 180.0 Median : 806.0 Median :109.0   
## Mean : 938 Mean : 235.2 Mean : 834.3 Mean :112.4   
## 3rd Qu.:1105 3rd Qu.: 284.0 3rd Qu.: 960.0 3rd Qu.:133.0   
## Max. :2214 Max. :1479.0 Max. :2683.0 Max. :340.0   
## PT08.S4.NO2. PT08.S5.O3. T RH AH   
## Min. : 551 Min. : 221 Min. :-1.9 Min. : 9.20 Min. :0.1847   
## 1st Qu.:1242 1st Qu.: 742 1st Qu.:12.0 1st Qu.:36.60 1st Qu.:0.7461   
## Median :1463 Median : 963 Median :17.8 Median :49.60 Median :0.9954   
## Mean :1457 Mean :1021 Mean :18.3 Mean :49.25 Mean :1.0244   
## 3rd Qu.:1662 3rd Qu.:1255 3rd Qu.:24.1 3rd Qu.:61.90 3rd Qu.:1.2962   
## Max. :2775 Max. :2523 Max. :44.6 Max. :88.70 Max. :2.2310

# adding date\_time column  
df\_named$date\_time <- df$date\_time

1. Explore columns

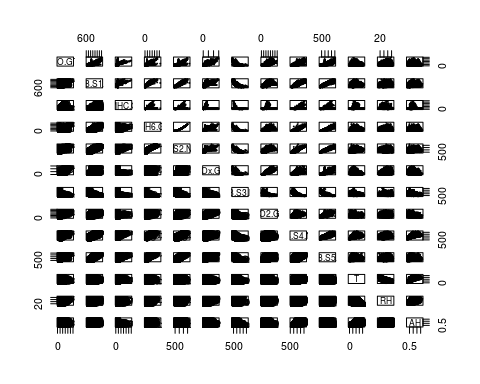
As we can see, probably it wasn`t a good idea to replace outliers with median, because there was to many of them. We also can see, that not all variables has normal distribution. This can bias the linear regression analysis

df\_named%>%  
 mutate(id=c(1:nrow(df\_named)))%>%  
 melt(measure.vars=c(1:13))->long  
  
ggplot(long, aes(value)) +   
 geom\_histogram(bins=40) +   
 facet\_wrap(~variable, scales = "free")



1. Let`s discovery cross-correlations

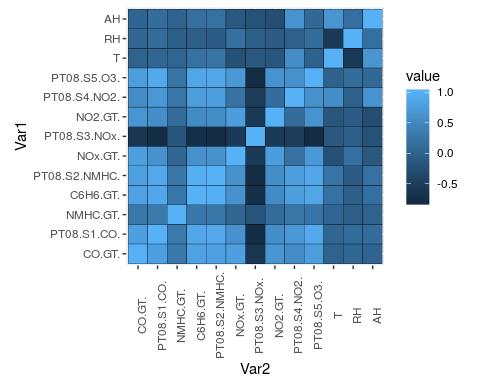
# plot correlations  
panel.points<-function(x,y)  
{  
 points(x,y,cex=.01)  
}  
  
pairs(df\_named[, 1:13], upper.panel = panel.points)



# correlation heatmap - more comportable way to find cross-correlations  
cor\_mtx <- (cor(df\_named[,1:13]))  
  
# all correlations in mtx  
cor\_mtx

## CO.GT. PT08.S1.CO. NMHC.GT. C6H6.GT. PT08.S2.NMHC.  
## CO.GT. 1.000000000 0.77694776 0.29105348 0.80885654 0.79595937  
## PT08.S1.CO. 0.776947756 1.00000000 0.30612962 0.88387075 0.89298861  
## NMHC.GT. 0.291053479 0.30612962 1.00000000 0.27042800 0.26807390  
## C6H6.GT. 0.808856543 0.88387075 0.27042800 1.00000000 0.98159631  
## PT08.S2.NMHC. 0.795959375 0.89298861 0.26807390 0.98159631 1.00000000  
## NOx.GT. 0.780462936 0.62254930 0.04885537 0.61614877 0.60617926  
## PT08.S3.NOx. -0.619318315 -0.77054354 -0.19077921 -0.73350297 -0.79579163  
## NO2.GT. 0.656001649 0.56344173 0.12100938 0.53331319 0.56193774  
## PT08.S4.NO2. 0.548481357 0.68236292 0.25768400 0.76457839 0.77696865  
## PT08.S5.O3. 0.763513032 0.89941732 0.22919013 0.86571073 0.88063309  
## T 0.006049053 0.04898505 0.03017121 0.19927329 0.24155572  
## RH 0.041137032 0.11440012 -0.04686398 -0.06181193 -0.09045007  
## AH 0.022863916 0.13572782 -0.01215017 0.16848094 0.18719567  
## NOx.GT. PT08.S3.NOx. NO2.GT. PT08.S4.NO2. PT08.S5.O3.  
## CO.GT. 0.78046294 -0.61931831 0.65600165 0.54848136 0.76351303  
## PT08.S1.CO. 0.62254930 -0.77054354 0.56344173 0.68236292 0.89941732  
## NMHC.GT. 0.04885537 -0.19077921 0.12100938 0.25768400 0.22919013  
## C6H6.GT. 0.61614877 -0.73350297 0.53331319 0.76457839 0.86571073  
## PT08.S2.NMHC. 0.60617926 -0.79579163 0.56193774 0.77696865 0.88063309  
## NOx.GT. 1.00000000 -0.57243986 0.76071574 0.20082479 0.69482257  
## PT08.S3.NOx. -0.57243986 1.00000000 -0.57418076 -0.53841242 -0.79533702  
## NO2.GT. 0.76071574 -0.57418076 1.00000000 0.13998573 0.63042540  
## PT08.S4.NO2. 0.20082479 -0.53841242 0.13998573 1.00000000 0.59076410  
## PT08.S5.O3. 0.69482257 -0.79533702 0.63042540 0.59076410 1.00000000  
## T -0.24570818 -0.14480191 -0.16938010 0.56118317 -0.02681366  
## RH 0.18393973 -0.05681926 -0.08213459 -0.03217158 0.12477593  
## AH -0.14828604 -0.23159695 -0.29820576 0.62951123 0.07115105  
## T RH AH  
## CO.GT. 0.006049053 0.04113703 0.02286392  
## PT08.S1.CO. 0.048985051 0.11440012 0.13572782  
## NMHC.GT. 0.030171212 -0.04686398 -0.01215017  
## C6H6.GT. 0.199273295 -0.06181193 0.16848094  
## PT08.S2.NMHC. 0.241555716 -0.09045007 0.18719567  
## NOx.GT. -0.245708185 0.18393973 -0.14828604  
## PT08.S3.NOx. -0.144801910 -0.05681926 -0.23159695  
## NO2.GT. -0.169380104 -0.08213459 -0.29820576  
## PT08.S4.NO2. 0.561183167 -0.03217158 0.62951123  
## PT08.S5.O3. -0.026813661 0.12477593 0.07115105  
## T 1.000000000 -0.57862533 0.65645249  
## RH -0.578625331 1.00000000 0.16788952  
## AH 0.656452492 0.16788952 1.00000000

# heatmap  
ggplot(data = melt(cor\_mtx), aes(Var2, Var1, fill = value))+  
 geom\_tile(color = "black")+  
 theme(axis.text.x = element\_text(angle = 90))+  
 coord\_fixed()



1. Linear regression models. Variable C6H6.GT. against all other variables.

fit <- lm(data = df\_named[,1:13], formula = C6H6.GT. ~. )   
  
# significant dependencies are marked with \*\*\* and \*\*  
summary(fit)

##   
## Call:  
## lm(formula = C6H6.GT. ~ ., data = df\_named[, 1:13])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.6701 -0.7221 -0.1927 0.5540 17.1756   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.041e+01 2.274e-01 -89.755 < 2e-16 \*\*\*  
## CO.GT. 2.359e-01 2.130e-02 11.078 < 2e-16 \*\*\*  
## PT08.S1.CO. 1.322e-03 1.611e-04 8.207 2.56e-16 \*\*\*  
## NMHC.GT. 2.873e-04 2.001e-04 1.436 0.15117   
## PT08.S2.NMHC. 2.943e-02 2.015e-04 146.050 < 2e-16 \*\*\*  
## NOx.GT. 1.712e-03 1.481e-04 11.560 < 2e-16 \*\*\*  
## PT08.S3.NOx. 4.113e-03 1.030e-04 39.927 < 2e-16 \*\*\*  
## NO2.GT. -9.082e-03 5.082e-04 -17.871 < 2e-16 \*\*\*  
## PT08.S4.NO2. -3.325e-04 1.281e-04 -2.595 0.00946 \*\*   
## PT08.S5.O3. -1.310e-04 8.934e-05 -1.466 0.14264   
## T -9.504e-02 5.150e-03 -18.454 < 2e-16 \*\*\*  
## RH -2.409e-02 1.988e-03 -12.115 < 2e-16 \*\*\*  
## AH 1.513e+00 9.656e-02 15.673 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.153 on 9344 degrees of freedom  
## Multiple R-squared: 0.9752, Adjusted R-squared: 0.9751   
## F-statistic: 3.059e+04 on 12 and 9344 DF, p-value: < 2.2e-16

1. Explore dependencies more carefully

We split data to train and test dataset

sample <- sample.int(n = nrow(df\_named), size = floor(.75\* nrow(df\_named)))  
  
train <- df\_named[sample,]  
nrow(train)

## [1] 7017

test <- df\_named[-sample,]  
nrow(test)

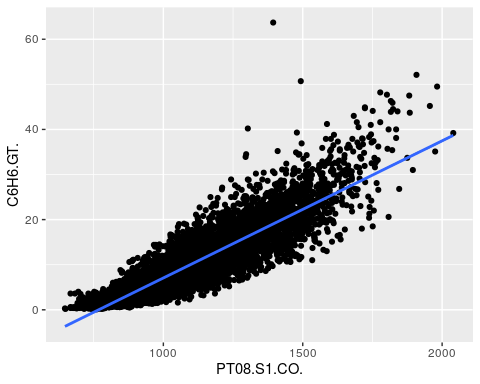
## [1] 2340

Check C6H6.GT. against PT08.S1.CO.

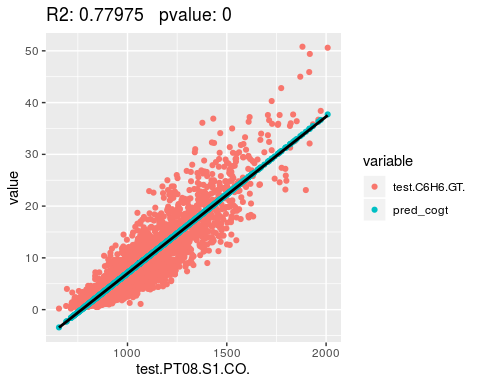
# model  
fit\_cogt <- lm(data = train[,1:13], formula = C6H6.GT. ~ PT08.S1.CO. )   
cogt <- summary(fit\_cogt)  
cogt

##   
## Call:  
## lm(formula = C6H6.GT. ~ PT08.S1.CO., data = train[, 1:13])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.300 -1.920 -0.230 1.807 44.659   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -23.356848 0.215721 -108.3 <2e-16 \*\*\*  
## PT08.S1.CO. 0.030415 0.000193 157.6 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.425 on 7015 degrees of freedom  
## Multiple R-squared: 0.7798, Adjusted R-squared: 0.7797   
## F-statistic: 2.484e+04 on 1 and 7015 DF, p-value: < 2.2e-16

# plot  
ggplot(train[,1:13], aes( PT08.S1.CO., C6H6.GT.))+  
 geom\_point()+  
 geom\_smooth(method='lm')



# predict test values  
pred\_cogt <- predict(fit\_cogt, test)  
  
# plot actual and predicted values  
test\_plot <- data.frame(test$PT08.S1.CO., test$C6H6.GT., pred\_cogt)  
  
qplot(test.PT08.S1.CO., value,   
 data = melt(test\_plot, measure.vars=c("test.C6H6.GT.", "pred\_cogt")),   
 colour=variable) +  
 geom\_smooth(method='lm', col='black')+  
 ggtitle(paste("R2:", round(cogt$adj.r.squared, 5), " pvalue:", round(cogt$coefficients[, 4], 5) ))

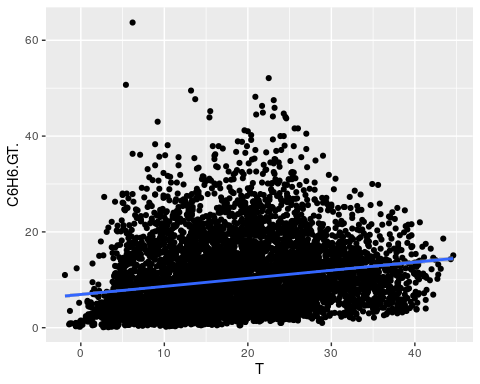


Another model: C6H6.GT. against T We can see, that linear regression probably found the nonexistent dependency due to nonlinear data

# model  
fit\_t <- lm(data = train[,1:13], formula = C6H6.GT. ~ T )   
t <- summary(fit\_t)  
t

##   
## Call:  
## lm(formula = C6H6.GT. ~ T, data = train[, 1:13])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10.805 -5.152 -1.737 3.212 55.715   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.941661 0.199431 34.81 <2e-16 \*\*\*  
## T 0.168298 0.009837 17.11 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.152 on 7015 degrees of freedom  
## Multiple R-squared: 0.04006, Adjusted R-squared: 0.03992   
## F-statistic: 292.7 on 1 and 7015 DF, p-value: < 2.2e-16

ggplot(train[,1:13], aes( T, C6H6.GT.))+  
 geom\_point()+  
 geom\_smooth(method='lm')



# predict test values based on the model  
pred\_t <- predict(fit\_t, test)  
  
t\_plot <- data.frame(test$T, test$C6H6.GT., pred\_t)  
  
# plot actual and predicted values  
qplot(test.T, value,   
 data = melt(t\_plot, measure.vars=c("test.C6H6.GT.", "pred\_t")),   
 colour=variable) +  
 geom\_smooth(method='lm', col='black')+  
 ggtitle(paste("R2:", round(t$adj.r.squared, 5), " pvalue:", round(t$coefficients[, 4], 5) ))

