STA141C Final Project

Abdallah Anees, Wanxin Hu, Stephanie Olivera, Hangyu Yue

The Dataset

In this analysis, we will by studying the Air Quality Data Set from the UCI Machine Learning Repository. This data was recorded in a significantly polluted area in an Italian City from March 2004 through February 2005.

The data set is made up of 9358 instances where each instance contains the hourly averaged responses from 5 metal oxide sensors embedded in a gas multisensor device. These average concentrations are denoted by PT08.S1, PT08.S2, PT08.S3, PT08.S4 and PT08.S5. The true chemical concentrations are also provided by a co-located reference certified analyzer. The chemicals of interest in this data set are CO, Non Metanic Hydrocarbons(NMHC), Benzene(C6H6), Total Nitrogen Oxides (NOx) and Nitrogen Dioxide (NO2). The first few observations of the dataset are shown below:

```
##         Date                Time CO(GT) PT08.S1(CO) NMHC(GT)  C6H6(GT)
## 1 2004-03-10 1899-12-31 18:00:00    2.6     1360.00      150 11.881723
## 2 2004-03-10 1899-12-31 19:00:00    2.0     1292.25      112  9.397165
## 3 2004-03-10 1899-12-31 20:00:00    2.2     1402.00       88  8.997817
##   PT08.S2(NMHC) NOx(GT) PT08.S3(NOx) NO2(GT) PT08.S4(NO2) PT08.S5(O3)    T
## 1       1045.50     166      1056.25     113      1692.00     1267.50 13.6
## 2        954.75     103      1173.75      92      1558.75      972.25 13.3
## 3        939.25     131      1140.00     114      1554.50     1074.00 11.9
##       RH        AH
## 1 48.875 0.7577538
## 2 47.700 0.7254874
## 3 53.975 0.7502391
```

Exploratory Data Analysis

Before beginning our analysis, we wanted to study the data to determine which information to consider in our statistical analysis.

Data Preprocessing

We found that this data set has many missing values. The missing values are tagged with the value -200. We first identify these missing values and replace them with `NA's`. We did this so that the -200 values would not affect our computations.

The next step in our data preprocessing is to add information about the month, day, and hour for each recording. To do this, we simply extract the month and day from the POSIX Date column and extract the hour from the POSIX Time Column. We then add these three new columns to our data frame using mutate().

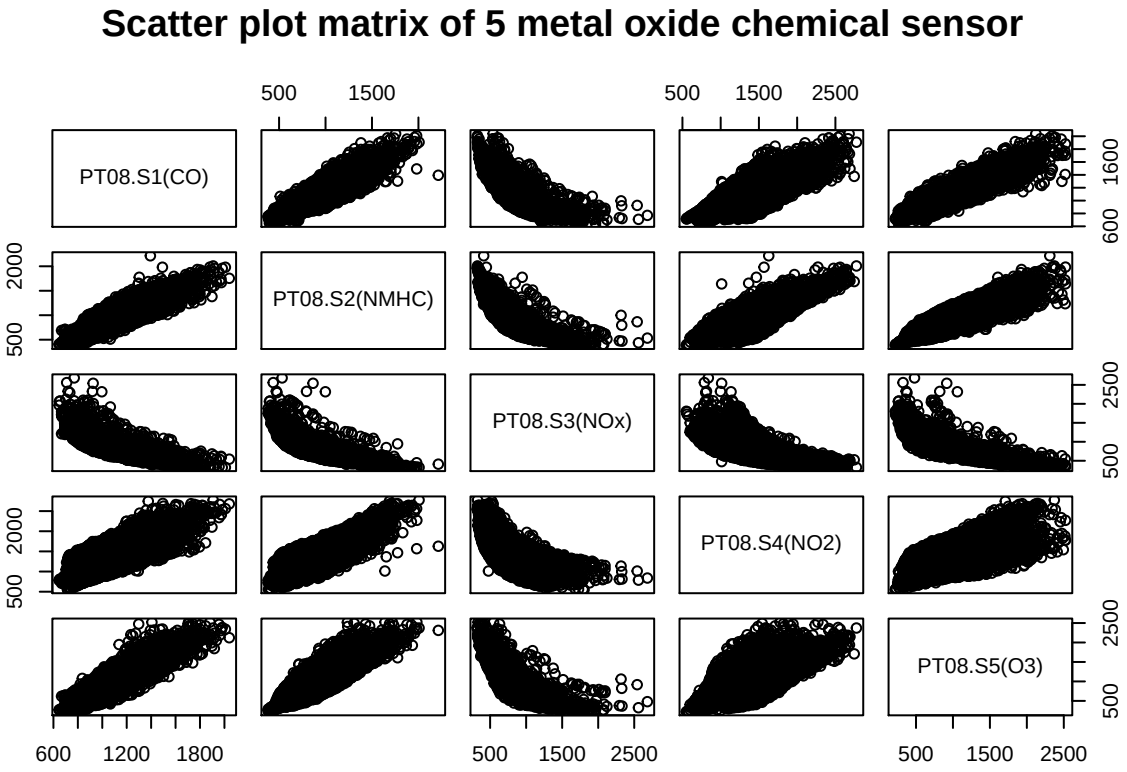
Our final processed data is shown below.

```
##         Date                Time CO(GT) PT08.S1(CO) NMHC(GT)  C6H6(GT)
## 1 2004-03-10 1899-12-31 18:00:00    2.6     1360.00      150 11.881723
## 2 2004-03-10 1899-12-31 19:00:00    2.0     1292.25      112  9.397165
## 3 2004-03-10 1899-12-31 20:00:00    2.2     1402.00       88  8.997817
##   PT08.S2(NMHC) NOx(GT) PT08.S3(NOx) NO2(GT) PT08.S4(NO2) PT08.S5(O3)    T
## 1       1045.50     166      1056.25     113      1692.00     1267.50 13.6
## 2        954.75     103      1173.75      92      1558.75      972.25 13.3
## 3        939.25     131      1140.00     114      1554.50     1074.00 11.9
##       RH        AH month day hour
## 1 48.875 0.7577538     3  10   18
## 2 47.700 0.7254874     3  10   19
## 3 53.975 0.7502391     3  10   20
```

Data Exploration

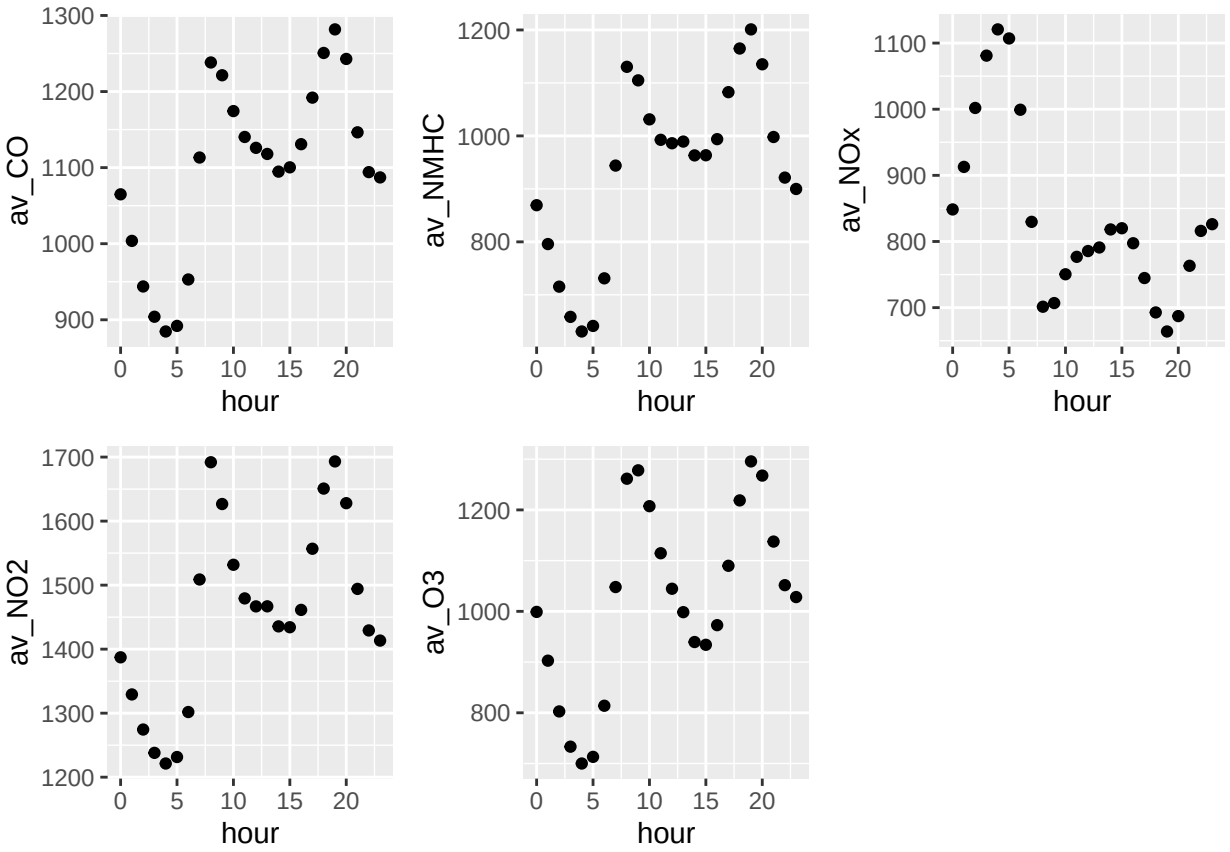
Identifying Correlations within the 5 Metal Oxide Chemical Sensors

We first created a scatter plot matrix of the 5 metal oxide chemical sensor readings to try to identify potential correlations among the chemical levels. From the scatter plot below, it is evident that a positive correlation exists between any two variables in CO, NMHC, NO2 and O3. We can also see that there is a negative correlation between NOx and all other variables CO, NMHC, NO2 and O3.



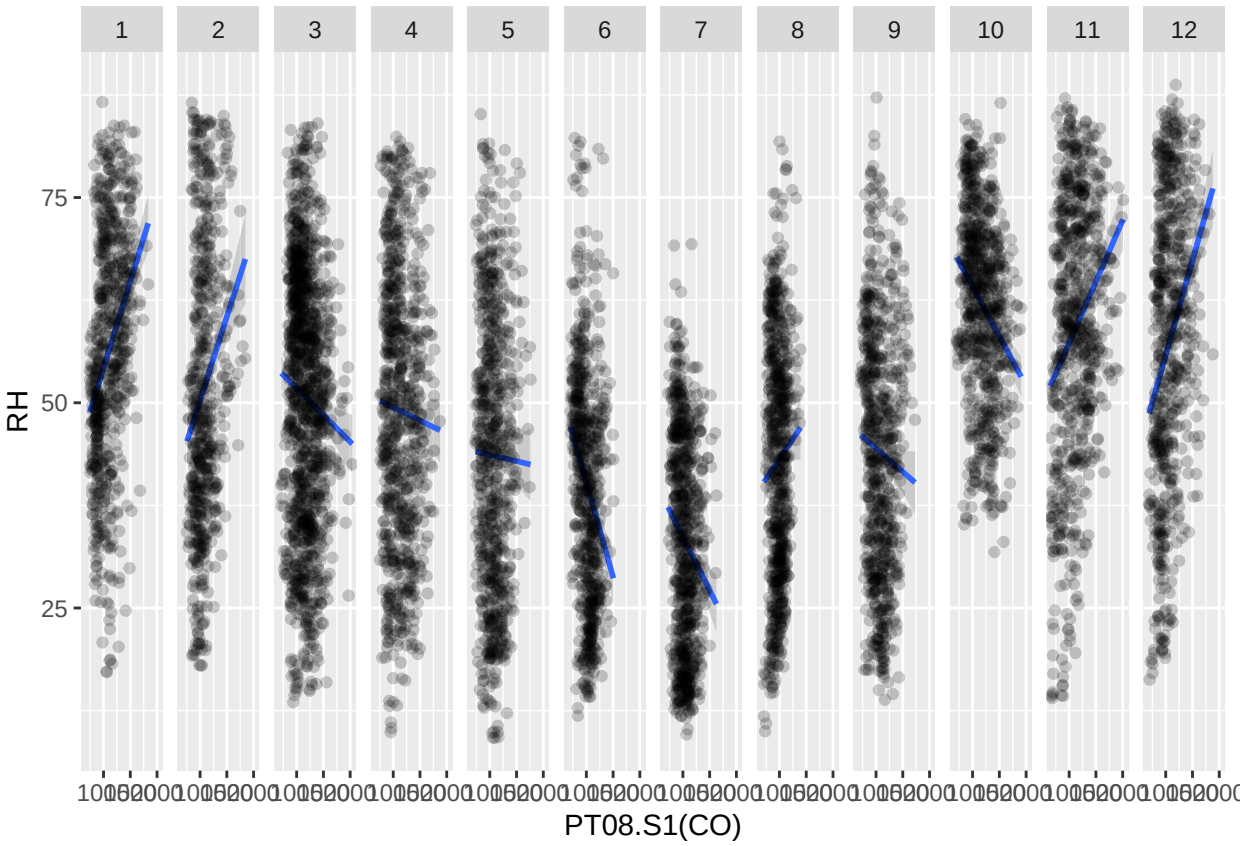
Computing Average Chemical Levels per Hour

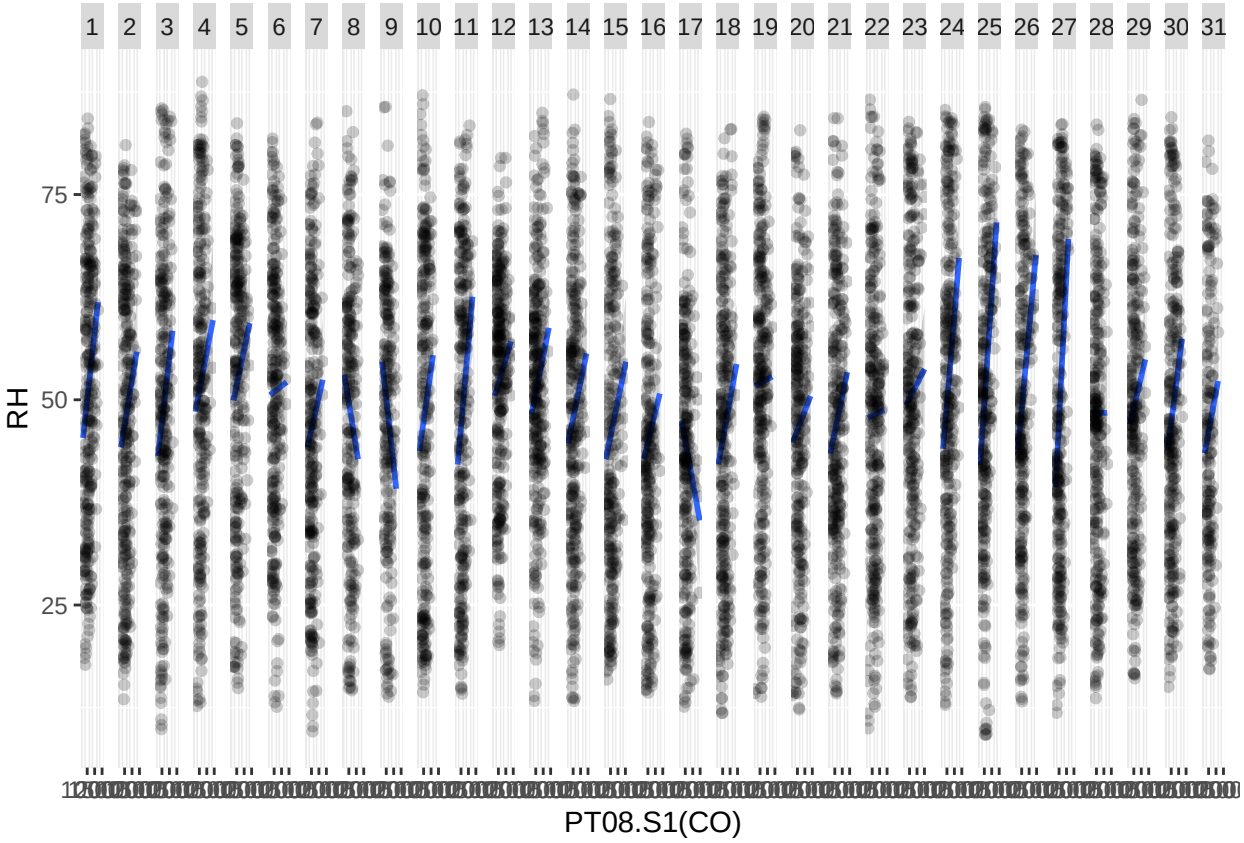
We determined the average level of each chemical per hour and created a table. We then used this table to plot the average level of each chemical for each hour. From the plots shown below, it is clear that there are certain hours of the day that tend to have the highest concentrations of chemicals CO, NMHC, NO2 and O3. For example, at hour 8, these chemicals are all at maximium average concentration while NOx is at the minimum average concentration.

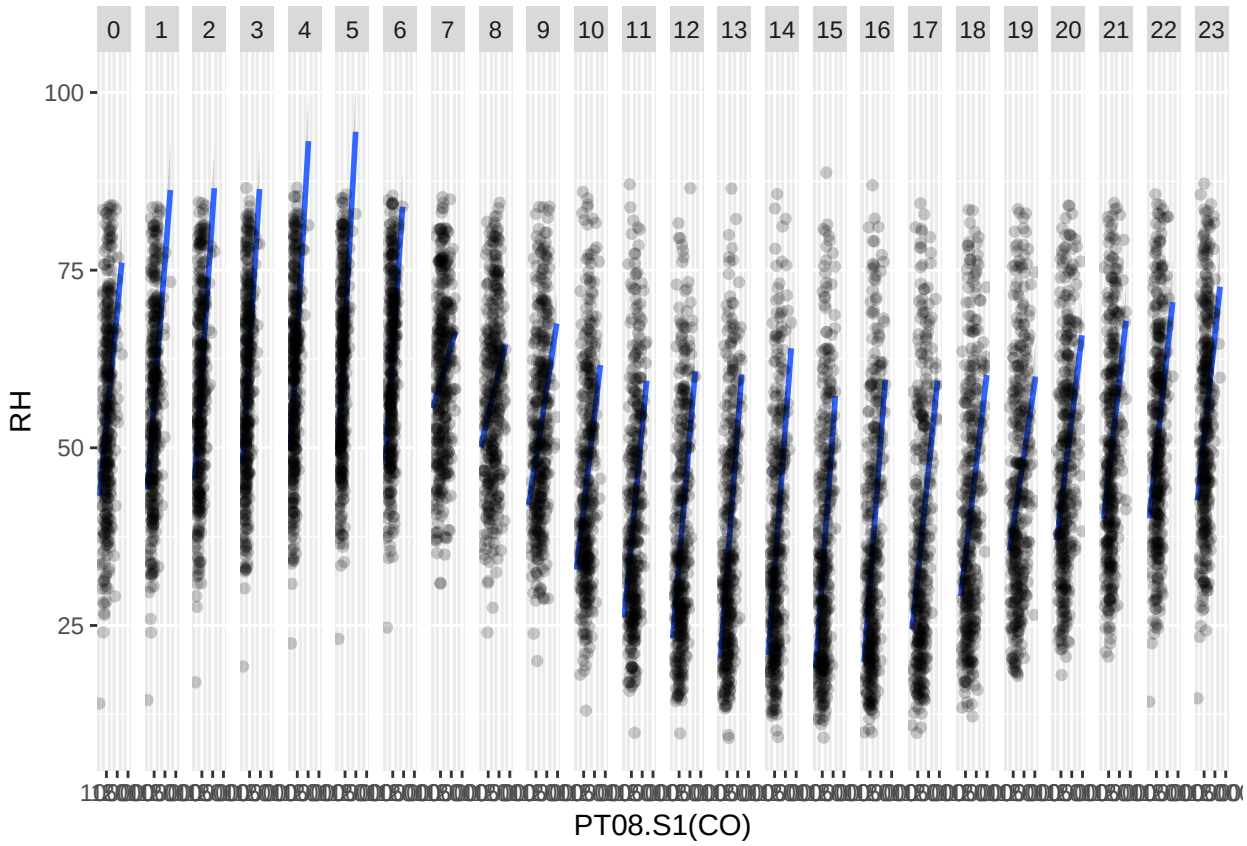


Correlation between CO and relative humidity across month, day and hour

To study the correlation between CO levels and relative humidty, we created the following plots of CO vs. Relative Humidity per month, per day, and per hour. It can be noted that there generally a strong positive correlation between CO and Relative Humidity levels per day and per hour.


1
2
3
4
5
6
7
8
9
10
11
12
75
50
25
RH
PT08.S1(CO)


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
75
50
25
RH
PT08.S1(CO)



Statistical Procedure: Multiple linear Regression

Let's say we are interested in predicting the concentration of CO. Then we can construct a multiple linear regression based on CO as follows:

```
##
## Call:
## lm(formula = `PT08.S1(CO)` ~ `PT08.S2(NMHC)` + `PT08.S3(NOx)` +
##     `PT08.S4(NO2)` + `PT08.S5(O3)` + T + RH + AH + month + day +
##     hour, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -427.89  -48.35   -2.78   42.65  356.17
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     387.107352  12.083005  32.037  < 2e-16 ***
## `PT08.S2(NMHC)`   0.363594   0.013146  27.659  < 2e-16 ***
## `PT08.S3(NOx)`   -0.018878   0.006562  -2.877 0.004026 **
## `PT08.S4(NO2)`    0.067855   0.008002   8.480  < 2e-16 ***
## `PT08.S5(O3)`     0.216720   0.005135  42.208  < 2e-16 ***
## T                 0.961936   0.321982   2.988 0.002820 **
## RH                2.114414   0.121984  17.333  < 2e-16 ***
## AH              -21.650975   6.197598  -3.493 0.000479 ***
## month            -9.258094   0.292289 -31.675  < 2e-16 ***
```

```
## day               0.098521   0.085797   1.148 0.250875
## hour              2.089063   0.127879  16.336  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 71.59 on 8980 degrees of freedom
##   (366 observations deleted due to missingness)
## Multiple R-squared:  0.8914, Adjusted R-squared:  0.8912
## F-statistic:  7368 on 10 and 8980 DF,  p-value: < 2.2e-16
```

From above, we observed that there are some insignificant explanatory variables, and we need to adjust our model. Simply remove insignificant explanatory variables T, AH and `PT08.S3(NOx)`. Our new model is shown below:

```
##
## Call:
## lm(formula = `PT08.S1(CO)` ~ `PT08.S2(NMHC)` + `PT08.S4(NO2)` +
##     `PT08.S5(O3)` + RH + month + hour, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -444.67  -48.06   -2.94   42.83  353.65
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     376.695973   4.428779   85.06   <2e-16 ***
## `PT08.S2(NMHC)`   0.385280   0.009704   39.70   <2e-16 ***
## `PT08.S4(NO2)`    0.059443   0.003823   15.55   <2e-16 ***
## `PT08.S5(O3)`     0.217523   0.004948   43.97   <2e-16 ***
## RH                1.789397   0.051263   34.91   <2e-16 ***
## month            -9.500933   0.229649  -41.37   <2e-16 ***
## hour              2.108946   0.127000   16.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 71.66 on 8984 degrees of freedom
##   (366 observations deleted due to missingness)
## Multiple R-squared:  0.8911, Adjusted R-squared:  0.891
## F-statistic: 1.225e+04 on 6 and 8984 DF,  p-value: < 2.2e-16
```

Bootstrap Procedure Applied to Multiple Linear Regression

In order to find the difference for each chemical based on temperature during the whole year, we need to apply the multiple linear regression model. Due the the large size of the data set, here we will use bag of little bootstrap with our multiple linear regression model:

We then begin our bag of little boot strap procedure by creating a sample_list of 10 samples where each sample contains about 900 observations. The dimensions of our sample_list samples is shown below.

```
##           X1  X2  X3  X4  X5  X6  X7  X8  X9 X10
## num_rows 901 964 917 921 941 982 925 951 925 929
## num_cols   6   6   6   6   6   6   6   6   6   6
```

We apply BLB to generate the following confidence intervals summarized in Table 1 and Table 2 below.

Table 1: Confidence Intervals for Linear Model with PT08.S1: CO as the Response

```
## # A tibble: 7 x 3
##   term                 low     high
##   <chr>              <dbl>    <dbl>
## 1 (Intercept)     346.     364.
## 2 `PT08.S2(NMHC)`   0.297    0.350
## 3 `PT08.S4(NO2)`    0.0451   0.0637
## 4 `PT08.S5(O3)`     0.240    0.266
## 5 day               0.0478   0.391
## 6 hour              2.07     2.61
## 7 RH                1.37     1.58
```

Table 2: Confidence Intervals for Linear Model with Temperature as the Response

```
## # A tibble: 13 x 3
##    term                 low      high
##    <chr>              <dbl>     <dbl>
##  1 (Intercept)      1.93    25.8
##  2 `C6H6(GT)`      -0.467    0.296
##  3 `CO(GT)`        -2.03    -0.492
##  4 `NMHC(GT)`      -0.00119  0.00290
##  5 `NO2(GT)`        0.00978  0.0490
##  6 `NOx(GT)`       -0.0303  -0.0180
##  7 `PT08.S1(CO)`   -0.00223  0.00543
##  8 `PT08.S2(NMHC)` -0.0238  -0.00973
##  9 `PT08.S3(NOx)`  -0.00832 -0.00204
## 10 `PT08.S4(NO2)`   0.0165   0.0341
## 11 `PT08.S5(O3)`   -0.00161  0.000437
## 12 hour            -0.0183   0.0247
## 13 RH              -0.332   -0.280
```

Results

Our goal here is to create confidence intervals for the coefficients of our Linear Regression model and to change to a different range for each of the 5 chemical compounds.

We used the Bag of Little Bootstraps (BLB) prodcedure, which incorporates features of both bootstrap and subsampling to find a computationally efficient means of the 5 chemical compound estimators in air pollution based on true hourly averaged concentration and the PT08 estimation.

The confidence intervals provide us with an upper and lower limit around our sample mean. Within this interval, it tells us the range of change of the chemical compound concentrations based on changing tempurature during the entire year. For example, in Table 2 the range of change for the chemical compound NO2 based true hourly averaged concentration is in between (-0.002428212 to 0.0009713231), and for the PT08. of NO2 the confidence interval is in between (0.027210390 to 0.0284737242). Here, we can see the reading for NO2 between the two concetration confidence intervals are different.

The confidence interval for the PT08. of NO2 is more accurate becuase there is no negative sign which means there is no zero in the confidence interval reading. The true hourly averaged concentration of NO2 confidence interval has a negative negative value for the lower limit.

Here, since we are measuring based on hourly change, the reading for the concentration is diffeerent for example at 3 am from the reading at 4 pm. The average change of difference in reading of the concentration is based on the activations for the 5 chimical compunds. For example, at 3 am the reading might be low since not a lot of cars and factories are working at that time, while at 3 pm and reading will likely be higher because cars are driven more often and factories are working at this time.

For example in Table 1, we test the confidence interval difference of PT08 for `PT08.S2(NMHC)`,`PT08.S4(NO2)`,`PT08.S5(O3)` based on the concentration of `PT08.S1(CO)`. From the table, we can see the change is different per hour, per day, and the Relative Humidity RH change per hour in air pollution. Here we can see the concentration rate change for the 3 cheimcal compounds are changing every hour, and their confidence interval was between 2.01977590 and 2.58940514. and the mean change for Relative Humidity RH every hour is between 1.97842172 and 2.05112489.

5. Code Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(readxl)
data <- read_excel("AirQualityUCI.xlsx")
data <- as.data.frame(data)
data %>% head(3)
# Replace -200 with `NA's`
data[data==-200] <- NA

## Add month, day, hour as additional columns
data <- data %>%
  mutate(
    month = as.double(format(data$Date, "%m")),
    day = as.double(format(data$Date, "%d")),
    hour = as.double(format(data$Time, "%H"))
)
data %>% head(3)
pairs(~`PT08.S1(CO)`+`PT08.S2(NMHC)`+`PT08.S3(NOx)`+
        `PT08.S4(NO2)`+`PT08.S5(O3)`, data = data, main = "Scatter plot matrix of 5 metal oxide chemical
av_data_h <- data %>%
  group_by(hour) %>%
  summarize(
    av_CO = mean(`PT08.S1(CO)`, na.rm = TRUE),
    av_NMHC = mean(`PT08.S2(NMHC)`, na.rm = TRUE),
    av_NOx = mean(`PT08.S3(NOx)`, na.rm = TRUE),
    av_NO2 = mean(`PT08.S4(NO2)`, na.rm = TRUE),
    av_O3 = mean(`PT08.S5(O3)`, na.rm = TRUE),
  )

## Display the 5 average metal oxide chemical sensors
library(gridExtra)
p1 <- ggplot(av_data_h, aes(x=hour, y=av_CO)) + geom_point()
p2 <- ggplot(av_data_h, aes(x=hour, y=av_NMHC)) + geom_point()
p3 <- ggplot(av_data_h, aes(x=hour, y=av_NOx)) + geom_point()
p4 <- ggplot(av_data_h, aes(x=hour, y=av_NO2)) + geom_point()
p5 <- ggplot(av_data_h, aes(x=hour, y=av_O3)) + geom_point()
grid.arrange(p1, p2, p3, p4, p5, nrow = 2)
co_pollution = ggplot(data, aes(`PT08.S1(CO)`, RH)) + geom_smooth(method = 'lm')
```

```r
co_pollution + geom_point(alpha = I(1/5)) + facet_grid(.~month)
co_pollution + geom_point(alpha = I(1/5)) + facet_grid(.~day)
co_pollution + geom_point(alpha = I(1/5)) + facet_grid(.~hour)
#### FINDING CONFIDENCE INTERVAL FOR THESE PREDICTIONS
fit1 <- lm(`PT08.S1(CO)` ~ `PT08.S2(NMHC)` +`PT08.S3(NOx)` +
             `PT08.S4(NO2)` + `PT08.S5(O3)`+`T`+ RH + AH + month + day + hour, data = data)
summary(fit1)
fit2 <- lm(`PT08.S1(CO)` ~ `PT08.S2(NMHC)` +
             `PT08.S4(NO2)` + `PT08.S5(O3)`+ RH + month + hour, data = data)
summary(fit2)
# import data
# data <- read_excel("~/Desktop/AirQualityUCI/AirQualityUCI.xlsx")
# data <- read_excel("AirQualityUCI.xlsx")

mydata = data
names(mydata)[1]='date'
names(mydata)[2]='time'
names(mydata)[3]='CO(GT)'
names(mydata)[4]='PT08.S1(CO)'
names(mydata)[5]='NMHC(GT)'
names(mydata)[6]='C6H6(GT)'
names(mydata)[7]='PT08.S2(NMHC)'
names(mydata)[8]='NOx(GT)'
names(mydata)[9]='PT08.S3(NOx)'
names(mydata)[10]='NO2(GT)'
names(mydata)[11]='PT08.S4(NO2)'
names(mydata)[12]='PT08.S5(O3)'
names(mydata)[13]='T'
names(mydata)[14]='RH'
names(mydata)[15]='AH'
mydata$V16<-NULL
mydata$V17<-NULL
mydata<-mydata[-c(1),]
# head(mydata)


library(rsample)
library(purrr)
library(broom)
library(dplyr)
set.seed(141)
n <- nrow(mydata)
m <- 10

mydata.df = data.frame(mydata$time, mydata$`PT08.S1(CO)`, mydata$`PT08.S2(NMHC)`, mydata$`PT08.S3(NOx)`
# head(mydata.df)
fit_1 <- lm(formula = mydata$T ~ mydata$`PT08.S1(CO)` + mydata$`PT08.S2(NMHC)` + mydata$`PT08.S3(NOx)`

subsample <- sample(seq_len(m), n, replace = TRUE)
sample_list <- mydata.df %>% split(subsample)
lapply(sample_list, dim) %>% as.data.frame(row.names=c("num_rows","num_cols"))
# Table 1
```

```
bootreg <- data %>%
  bootstraps(1000) %>%
  pull(splits) %>%
  map_dfr(~ {
    train_data <- analysis(.)
    lm(`PT08.S1(CO)` ~ `PT08.S2(NMHC)` +
             `PT08.S4(NO2)` + `PT08.S5(O3)`+ RH + day + hour, data = train_data) %>%
  tidy()
  })

summarize = dplyr::summarize

bootreg %>%
  group_by(term) %>%
  summarize(low=quantile(estimate, .025),
            high=quantile(estimate, .975))

#Table 2

bootreg = data %>%
  bootstraps(1000) %>%
  pull(splits) %>%
  map_dfr(~ {
    train_data <- analysis(.)
    lm(T ~ `PT08.S1(CO)` + `PT08.S2(NMHC)` + `PT08.S3(NOx)` + `PT08.S4(NO2)` + `PT08.S5(O3)` + `CO(GT)`
  tidy()
  })

summarize = dplyr::summarize

bootreg %>%
  group_by(term) %>%
  summarize(low=quantile(estimate, .025),
            high=quantile(estimate, .975))
```