STAT 331 Final Project

Maxine, Estella, Judy, Weiwei

04/12/2021

1. Summary

A maximum of 200 words describing the objective of the report, an overview of the statistical analysis, and summary of the main results.

2. Objective

The goal of this project is to analyze the pollutants.csv data and write a report on your analysis. The specific goals of your analysis are up to you to decide.

3. Exploratory Data Analysis

Conduct exploratory data analyses: report summary statistics, visualize data (histograms, scatter plots, etc.). Report on any interesting findings and comment on how these inform the rest of your analysis.

can use this as a tutorial https://r4ds.had.co.nz/exploratory-data-analysis.html

Import dataset

```r
# CHANGE ABSOLUTE PATH

# setwd("~/Desktop/stat341/R331project/data")
setwd("~/School/4A/STAT 331/R331project/data")
# setwd("~/Desktop/R331project/data")

pollutants_raw <- read.csv("pollutants.csv", header = TRUE)
names(pollutants_raw)
```

```
##  [1] "X"                "length"           "POP_PCB1"         "POP_PCB2"        
##  [5] "POP_PCB3"         "POP_PCB4"         "POP_PCB5"         "POP_PCB6"        
##  [9] "POP_PCB7"         "POP_PCB8"         "POP_PCB9"         "POP_PCB10"       
## [13] "POP_PCB11"        "POP_dioxin1"      "POP_dioxin2"      "POP_dioxin3"     
## [17] "POP_furan1"       "POP_furan2"       "POP_furan3"       "POP_furan4"      
## [21] "whitecell_count" "lymphocyte_pct"   "monocyte_pct"     "eosinophils_pct" 
## [25] "basophils_pct"    "neutrophils_pct" "BMI"              "edu_cat"         
## [29] "race_cat"         "male"             "ageyrs"           "yrssmoke"        
## [33] "smokenow"         "ln_lbxcot"
```

Note that "edu_cat", "race_cat", "male", "smokenow" are categorical data and X is the index column.

```r
# Mxn's work
# clean the pollutants dataframe
pollutants <- subset(pollutants_raw , select = -X)
```

```r
# deal with categorical data

# 1 = Less Than 9th Grade or 9-11th Grade (Includes 12th grade with no diploma)
# 2 = High School Grad/GED or Equivalent
# 3 = Some College or AA degree
# 4 = College Graduate
edu_factor=factor(pollutants$edu_cat)

# 1 = Other Race (Including Multi-Racial);
# 2 = Mexican American;
# 3 = Non-Hispanic Black;
# 4 = Non-Hispanic White
race_factor=factor(pollutants$race_cat,
                   labels = c("Other", "Mexican", "Black", "White"))

# 0 = does not currently smoke;
# 1 = currently smokes
smoke_factor=factor(pollutants$smokenow, labels = c("Non-Smoker", "Smoker"))

# 0 = female, 1 = male
gender_factor=factor(pollutants$male, labels = c("female", "male"))


pollutants$edu_cat = edu_factor
pollutants$race_cat = race_factor
pollutants$smokenow = smoke_factor
pollutants$male = gender_factor
```

Get the names of covariates after we have performed some cleaning on the data

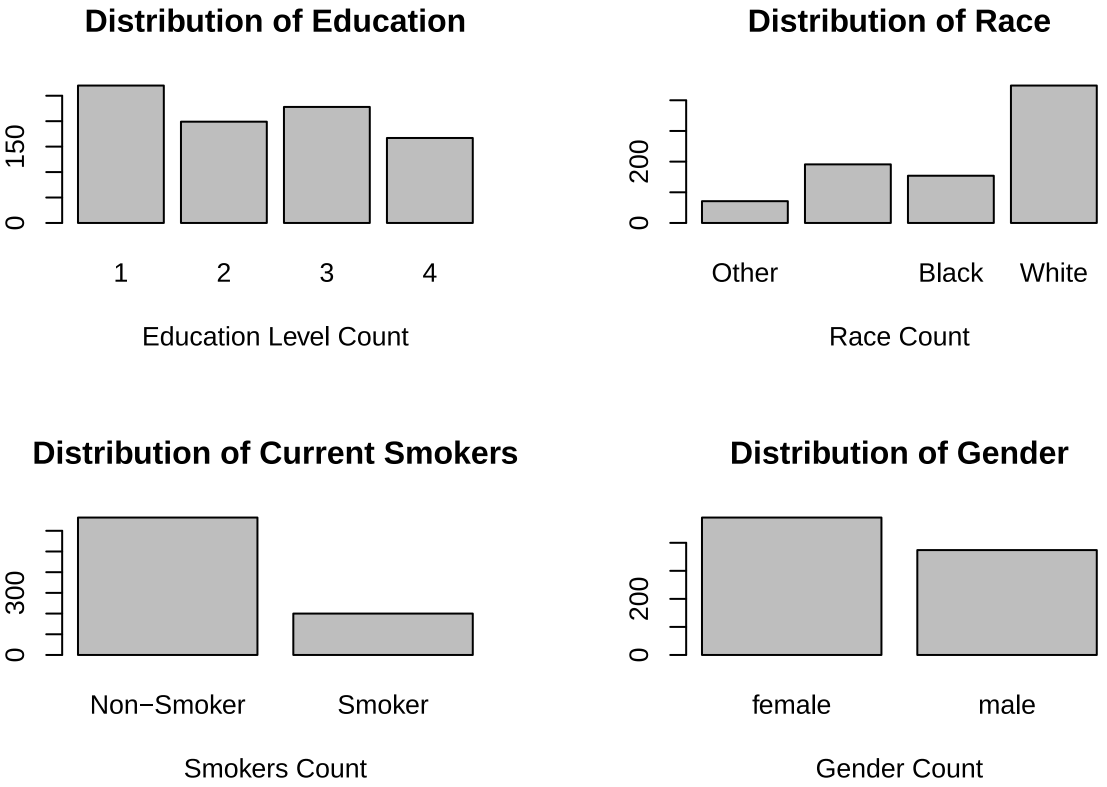
```r
names(pollutants)
```

```
##  [1] "length"          "POP_PCB1"        "POP_PCB2"        "POP_PCB3"
##  [5] "POP_PCB4"        "POP_PCB5"        "POP_PCB6"        "POP_PCB7"
##  [9] "POP_PCB8"        "POP_PCB9"        "POP_PCB10"       "POP_PCB11"
## [13] "POP_dioxin1"     "POP_dioxin2"     "POP_dioxin3"     "POP_furan1"
## [17] "POP_furan2"      "POP_furan3"      "POP_furan4"      "whitecell_count"
## [21] "lymphocyte_pct"  "monocyte_pct"    "eosinophils_pct" "basophils_pct"
## [25] "neutrophils_pct" "BMI"             "edu_cat"         "race_cat"
## [29] "male"            "ageyrs"          "yrssmoke"        "smokenow"
## [33] "ln_lbxcot"
```

Data Distribution

We investigate the distribution of covariates from the supplied data.

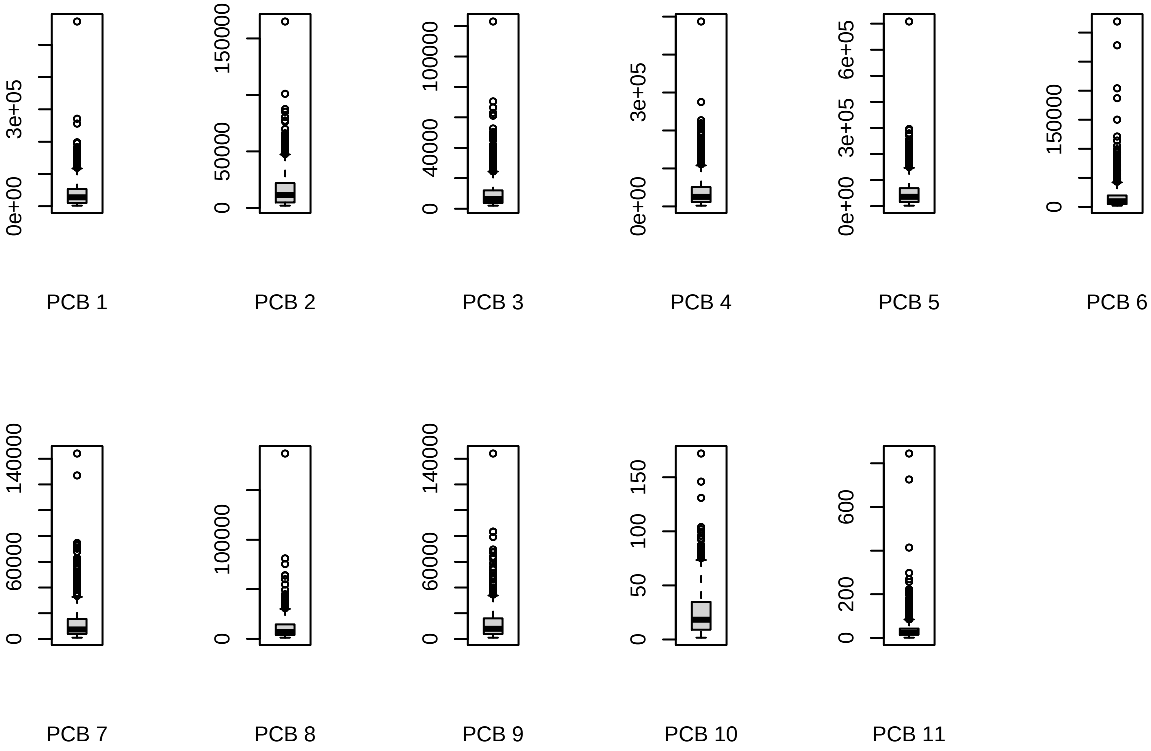
```r
# Mxn's work
# put bargraphs for categorical data onto one picture
par(mfrow=c(2,2))

plot(edu_factor,
     main="Distribution of Education",
     xlab="Education Level Count")

plot(race_factor,
     main="Distribution of Race",
     xlab="Race Count")

plot(smoke_factor,
     main="Distribution of Current Smokers",
     xlab="Smokers Count")

plot(gender_factor,
     main="Distribution of Gender",
     xlab="Gender Count")
```



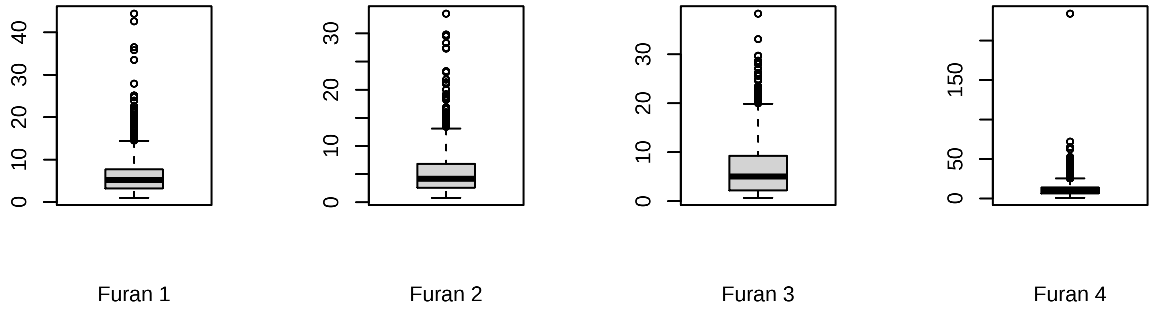
We see that we have more data about non-smokers than smokers and white people than other races. There are more entries for lower-education than higher, and more female than male.However, the distribution of gender and education are relatively close.

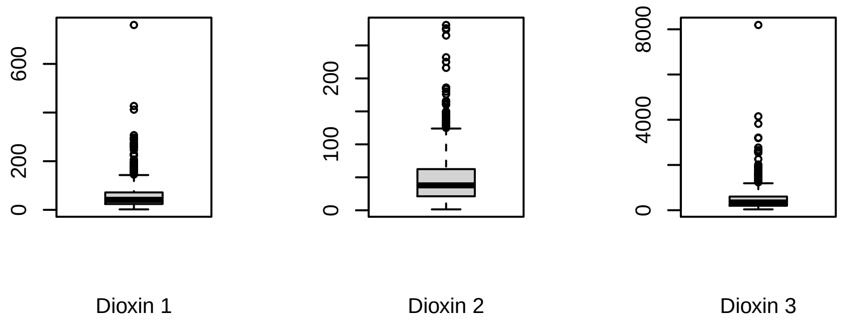
```
# Mxn's work
# PCB 1-6
par(mfrow=c(2,6))
boxplot(pollutants[, 2], xlab="PCB 1")
boxplot(pollutants[, 3], xlab="PCB 2")
boxplot(pollutants[, 4], xlab="PCB 3")
boxplot(pollutants[, 5], xlab="PCB 4")
boxplot(pollutants[, 6], xlab="PCB 5")
boxplot(pollutants[, 7], xlab="PCB 6")
boxplot(pollutants[, 8], xlab="PCB 7")
boxplot(pollutants[, 9], xlab="PCB 8")
boxplot(pollutants[, 10], xlab="PCB 9")
boxplot(pollutants[, 11], xlab="PCB 10")
boxplot(pollutants[, 12], xlab="PCB 11")
```

```
# Dioxin
par(mfrow=c(2,4))
```

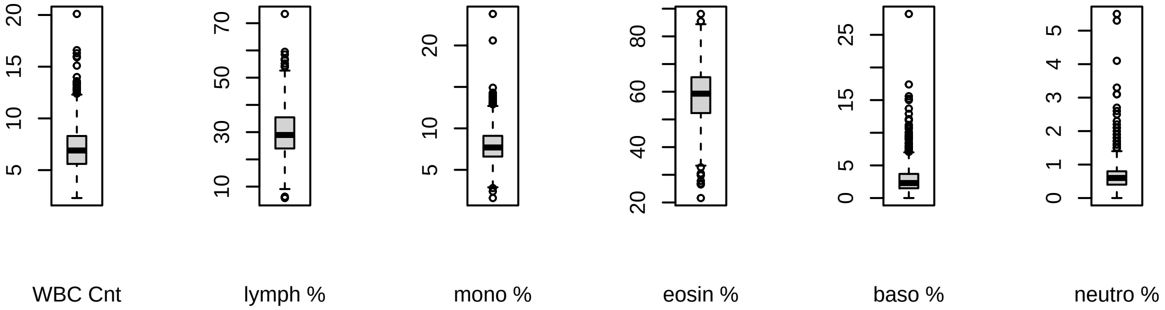


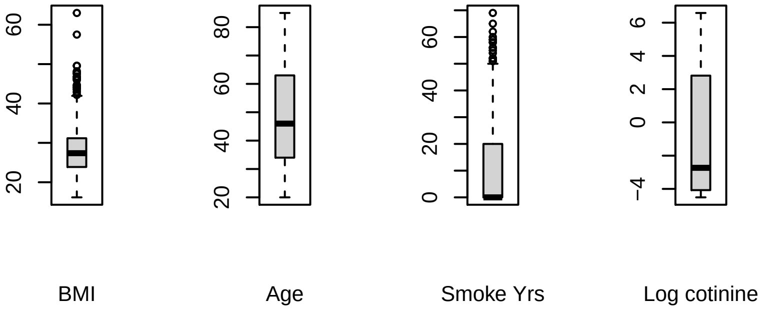
```
boxplot(pollutants[, 16], xlab="Furan 1")
boxplot(pollutants[, 17], xlab="Furan 2")
boxplot(pollutants[, 18], xlab="Furan 3")
boxplot(pollutants[, 19], xlab="Furan 4")
# Furan
boxplot(pollutants[, 13], xlab="Dioxin 1")
boxplot(pollutants[, 14], xlab="Dioxin 2")
boxplot(pollutants[, 15], xlab="Dioxin 3")

# white blood cells and concentrations
par(mfrow=c(2,6))
```





```
boxplot(pollutants[, 20], xlab="WBC Cnt")
boxplot(pollutants[, 21], xlab="lymph %")
boxplot(pollutants[, 22], xlab="mono %")
boxplot(pollutants[, 23], xlab="eosin %")
boxplot(pollutants[, 24], xlab="baso %")
boxplot(pollutants[, 25], xlab="neutro %")
# others
boxplot(pollutants[, 26], xlab="BMI")
boxplot(pollutants[, 30], xlab="Age")
boxplot(pollutants[, 31], xlab="Smoke Yrs")
boxplot(pollutants[, 33], xlab="Log cotinine")
```





We see that there are some extreme outliers in some concentration of PCBs, Dioxins, and Furan. The maximum values are sometimes over double the magnitude of the second largest.

However with a little investigation in @ref(#outlier-entries), we see that outliers for PCB values mostly came from one observation.

```
pollutants[436, 3:12]
```

```
##     POP_PCB2 POP_PCB3 POP_PCB4 POP_PCB5 POP_PCB6 POP_PCB7 POP_PCB8 POP_PCB9
## 436   165000   123000   487000   708000   319000   127000   187000   144000
##     POP_PCB10 POP_PCB11
## 436       131       137
```

Similarly, the most extreme outliers for different types of Dioxin and Furan also came from the same entry of data:

- Entry 285 contain the highest value for Dioxin 1 and 3, which are extreme outliers as we can see from the boxplots
- Entry 559 contain the highest value for Furan 2 and 4, where Furan 4 has an extreme outlier

Other covariates do not have a common entry that contribute to the outliers.

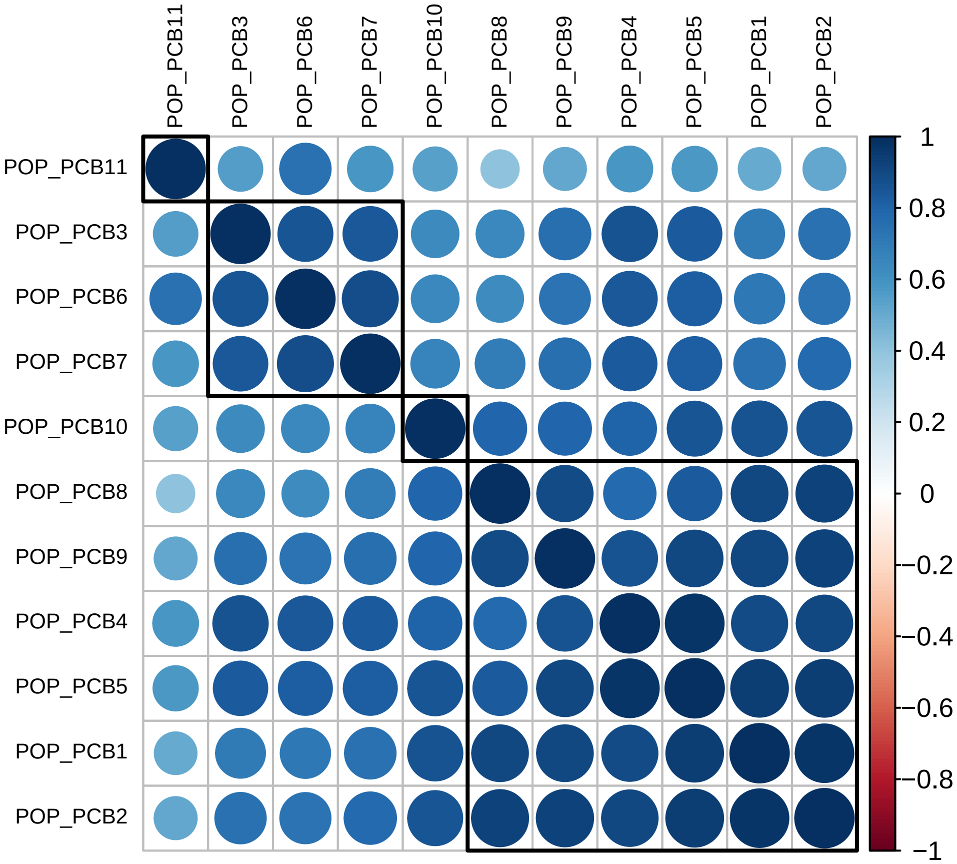
Multicolinearity

Correlation among PCB Concentrations

```
# Estella's work 1
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(ggplot2)

POP_PCB = c("POP_PCB1", "POP_PCB2","POP_PCB3", "POP_PCB4",
            "POP_PCB5", "POP_PCB6","POP_PCB7", "POP_PCB8",
            "POP_PCB9", "POP_PCB10", "POP_PCB11")
POP_PCB_data <- pollutants [, POP_PCB]
cc = cor(POP_PCB_data , method = "spearman")

# cluster my POP_PCB so that those with similar patterns
# of correlation coefficients are closer together.
# https://jkzorz.github.io/2019/06/11/Correlation-heatmaps.html
corrplot(cc, tl.col = "black", order = "hclust", hclust.method = "average",
         addrect = 4, tl.cex = 0.7)
```



Correlation among PCB Concentrations

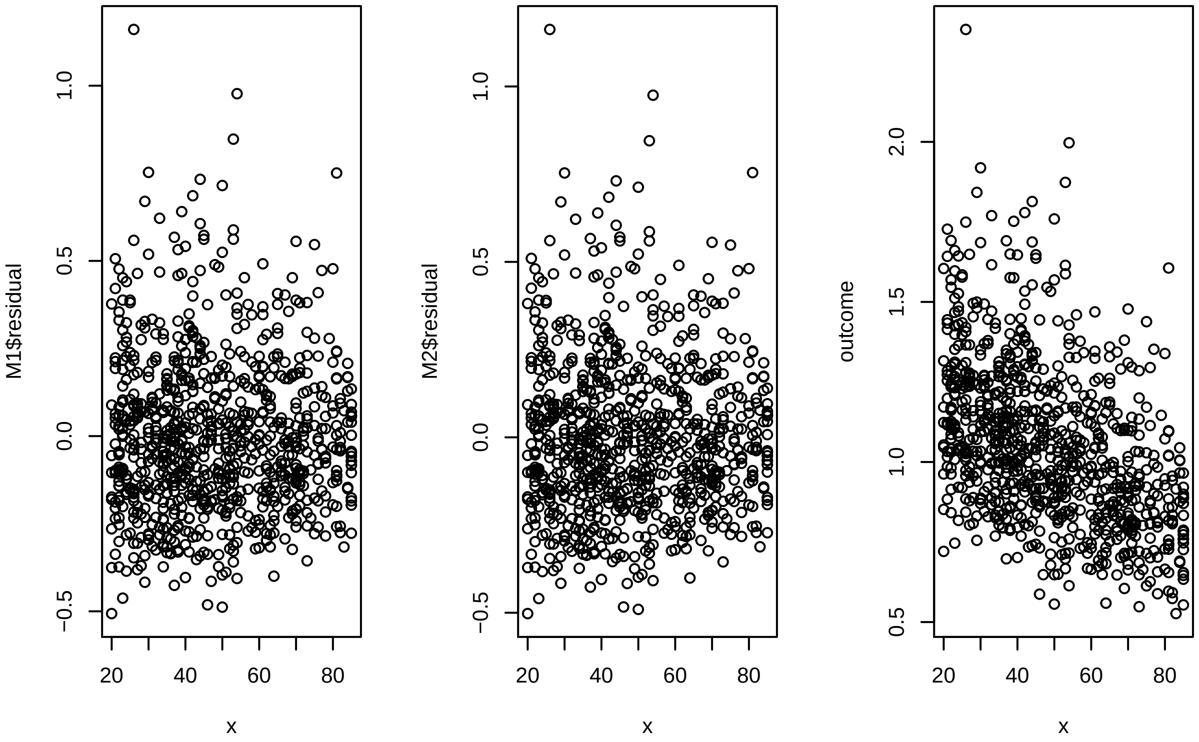
4. Methods:

Describe your statistical analysis: What is your model? Did you use any transformations or extensions of the basic multiple linear regression model? How did you select a model? Does the model fit the data well? Are the necessary assumptions met? Be sure to explain and justify your decisions.

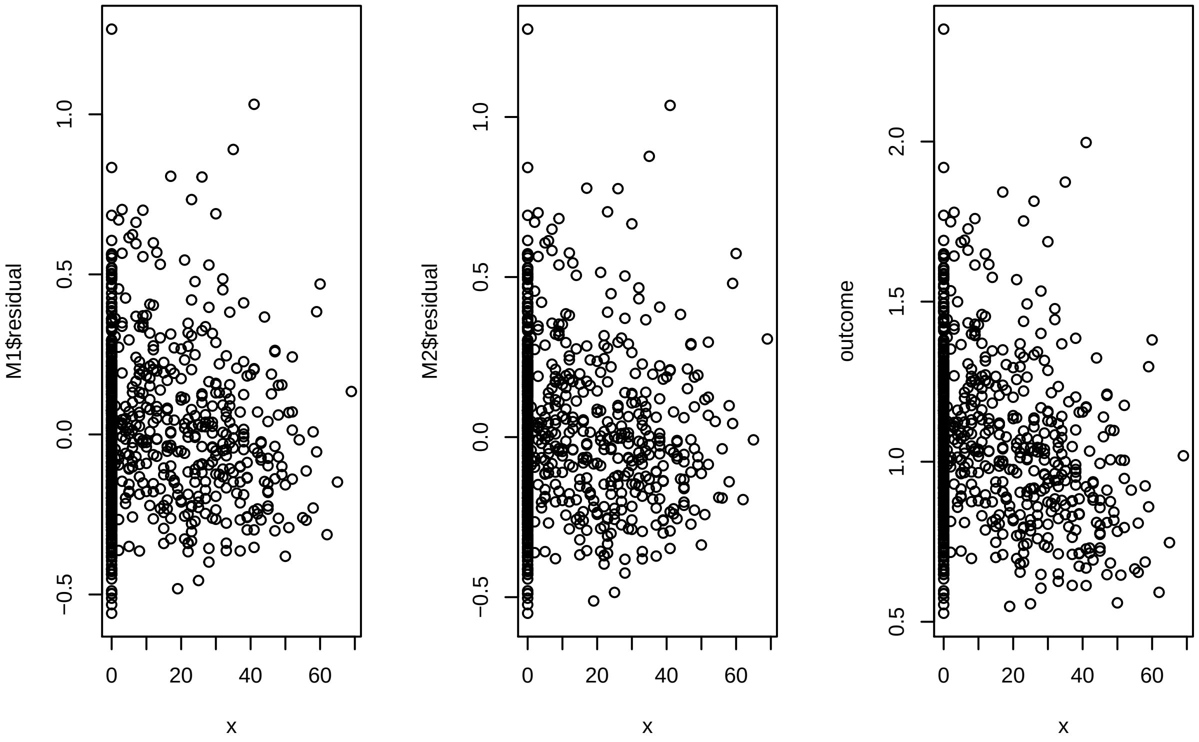
```
train_data <- pollutants[1:600,]
test_data <- pollutants[601:nrow(pollutants),]
```

```r
# Judy's work Part 1
# testing non-linearity in SLR
# if for any covariate, residual vs x for M1 has a pattern and
# residual vs x for M2 seems random, then y has a nonlinear
# relationship with with x.
# M1: fitting y to x
# M2: fitting y to x^2

par(mfrow=c(1, 3))
outcome <- pollutants$length
check <- function(x) {
  M1 <- lm(outcome ~ x)
  print(paste("residual for M1: ", sigma(M1)))
  M2 <- lm(outcome ~ x + I(x^2))
  print(paste("residual for M2: ", sigma(M2)))
  plot(x, M1$residual)
  plot(x, M2$residual)
  plot(x, outcome)
}

list <- list(pollutants$ageyrs, pollutants$yrssmoke,
             pollutants$BMI, pollutants$ln_lbxcot,
             pollutants$whitecell_count, pollutants$lymphocyte_pct,
             pollutants$monocyte_pct, pollutants$eosinophils_pct,
             pollutants$basophils_pct, pollutants$neutrophils_pct)
for (column in list) {
  check(column)
}
```

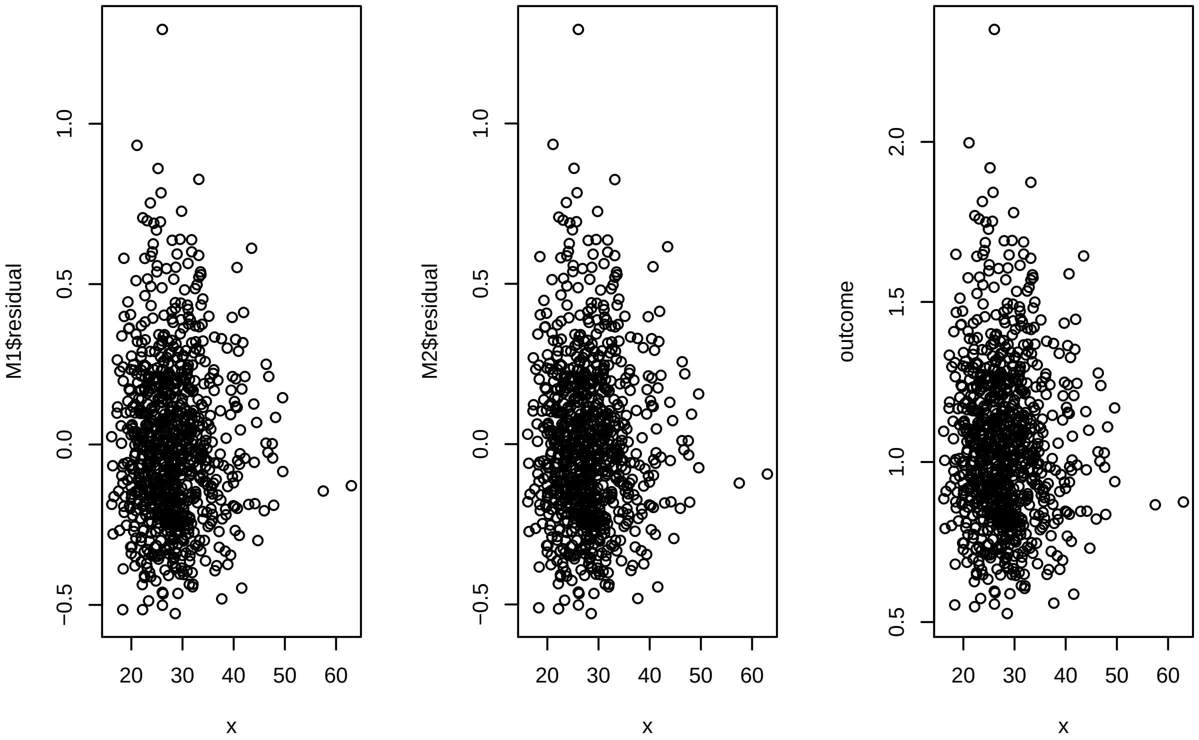
```
## [1] "residual for M1:  0.224172364185412"
## [1] "residual for M2:  0.22429269961392"
```



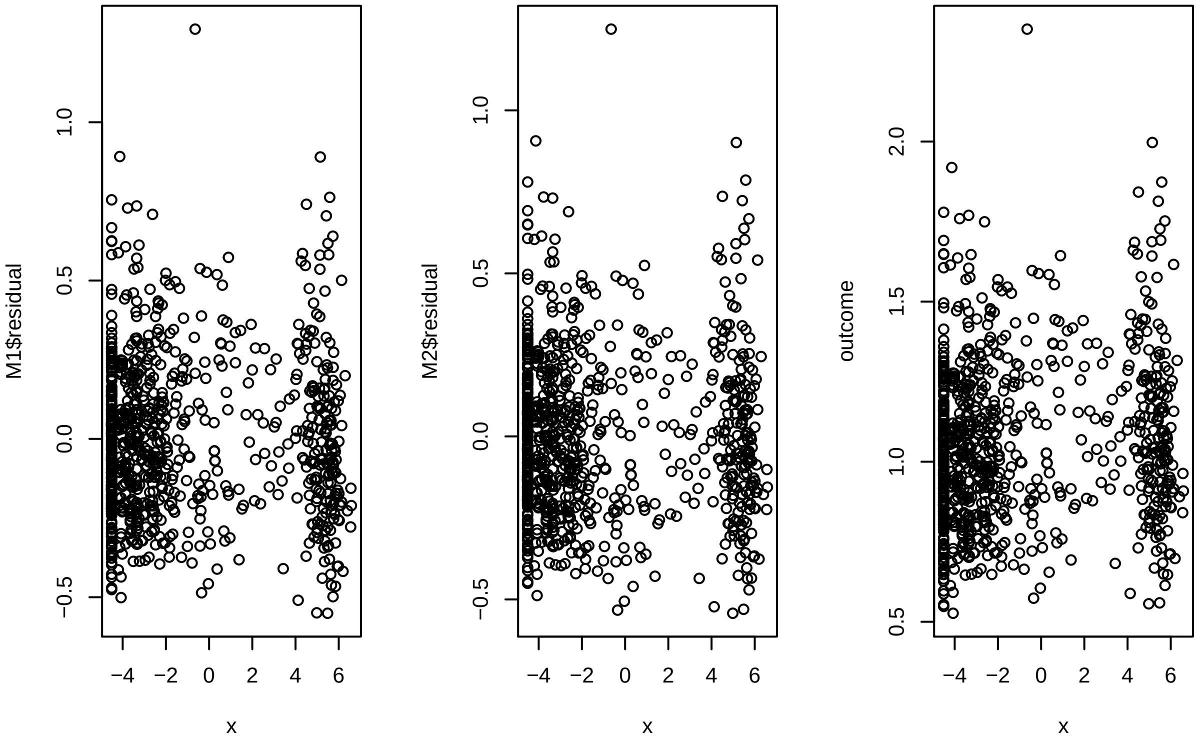
```
## [1] "residual for M1:  0.246320733146214"
## [1] "residual for M2:  0.245622720856213"
```



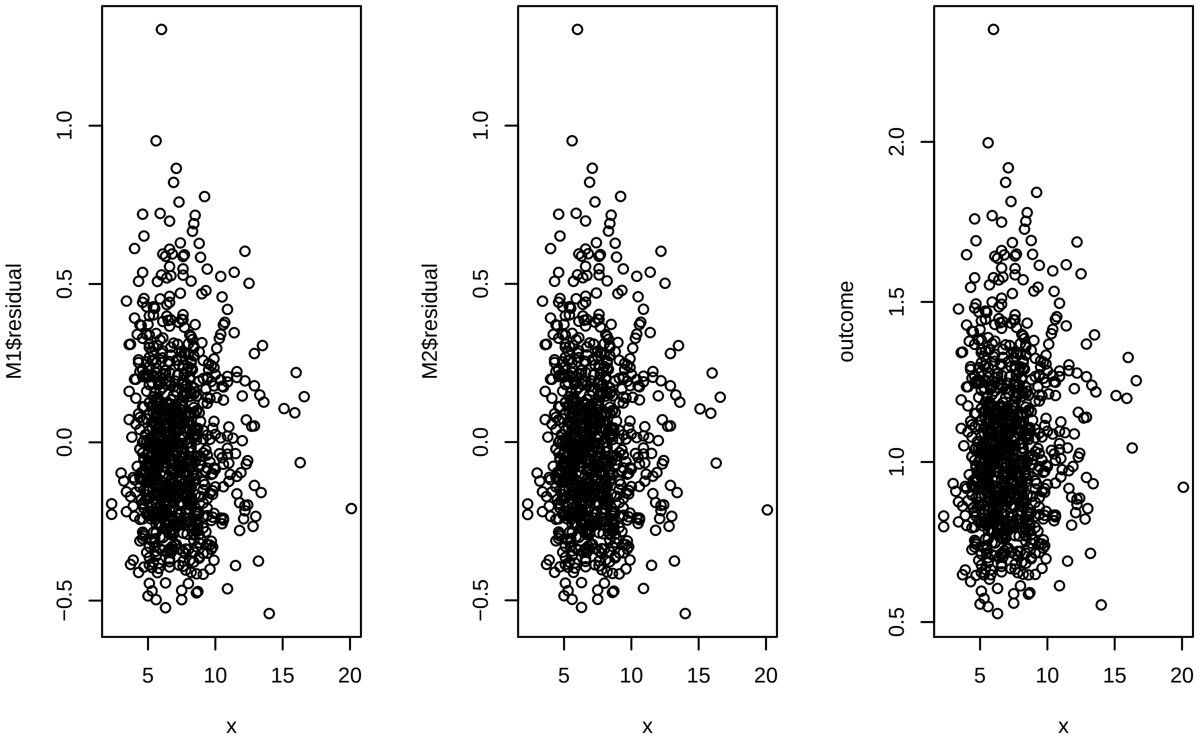
```
## [1] "residual for M1:  0.250228706427173"
## [1] "residual for M2:  0.25036248052387"
```



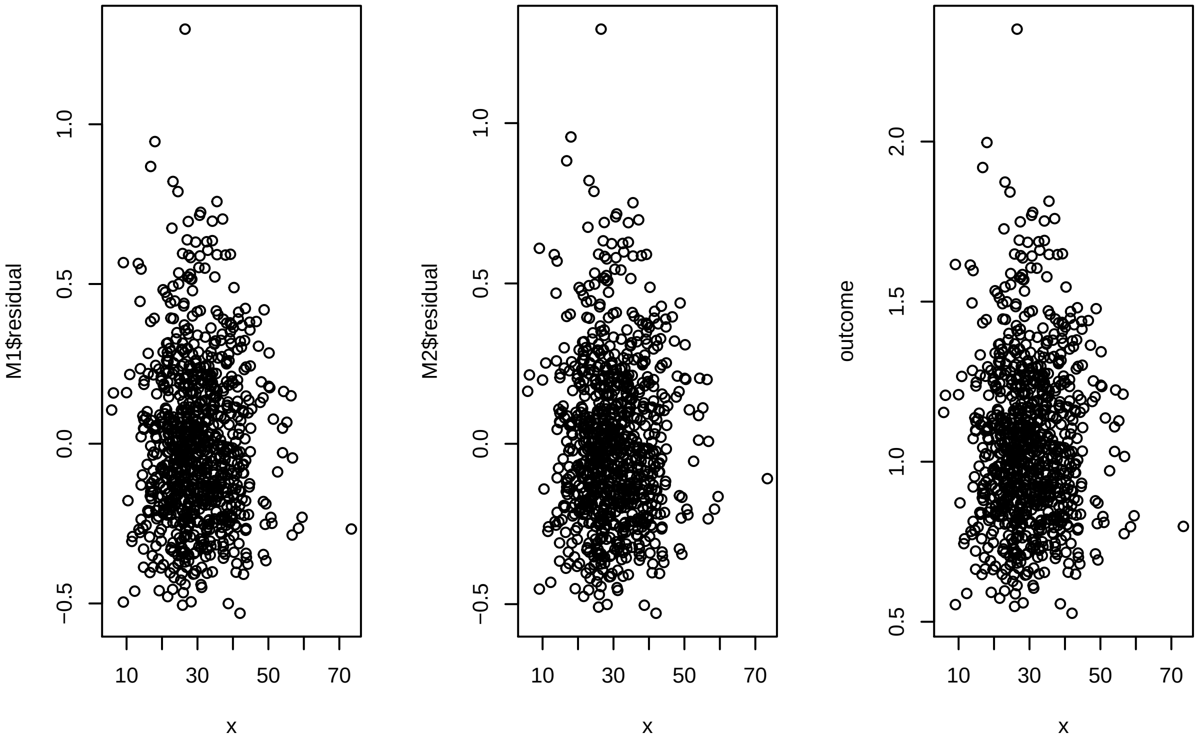
```
## [1] "residual for M1:  0.248212063673837"
## [1] "residual for M2:  0.24710732733351"
```



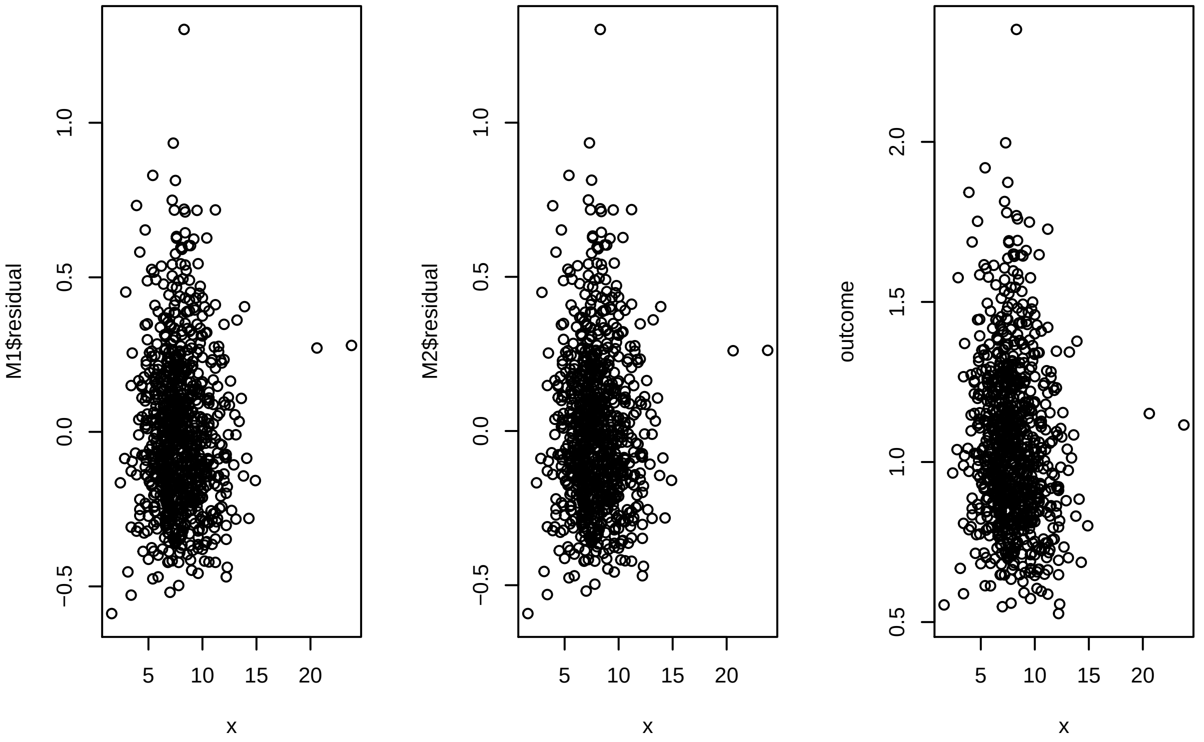
```
## [1] "residual for M1:  0.250065445847753"
## [1] "residual for M2:  0.250210403543218"
```



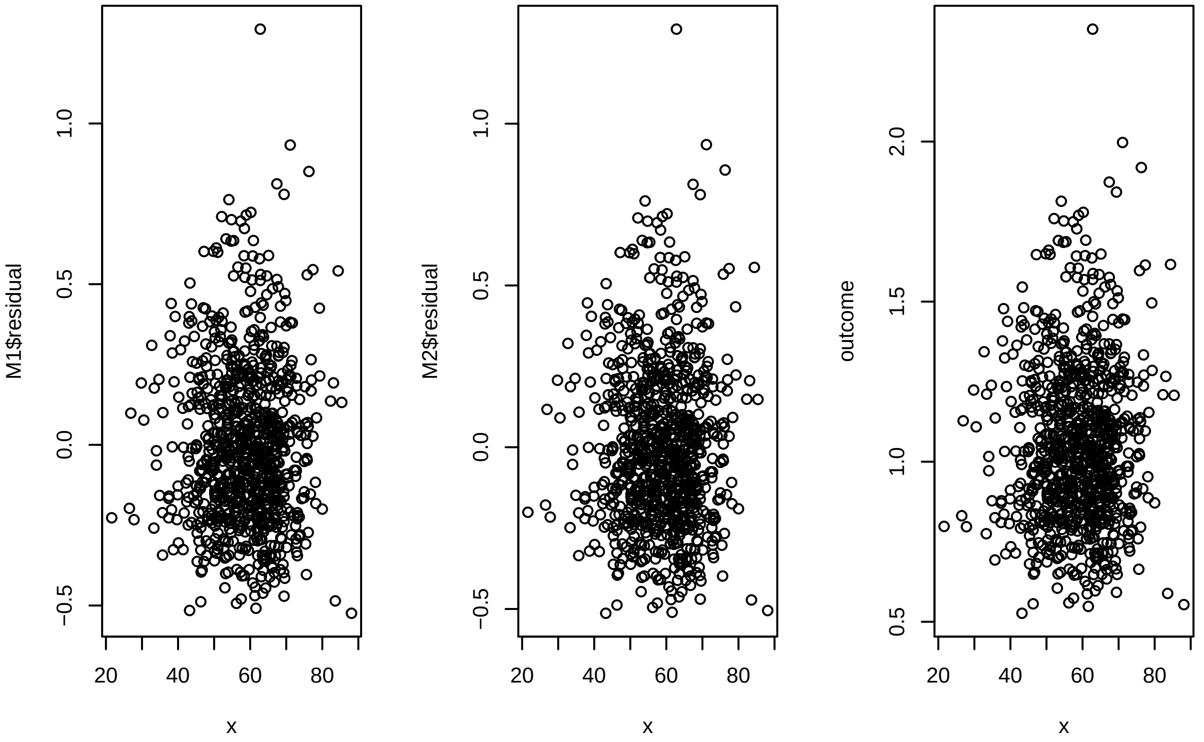
```
## [1] "residual for M1:  0.250373616826691"
## [1] "residual for M2:  0.250255208638358"
```



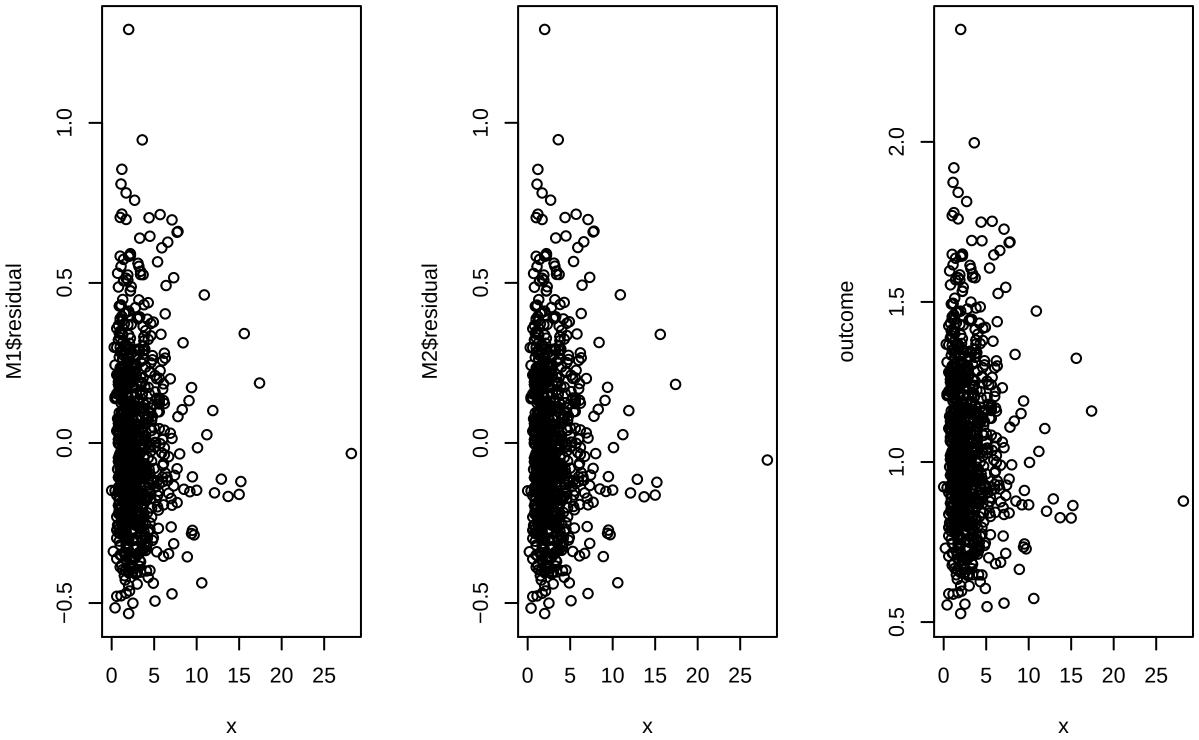
```
## [1] "residual for M1:  0.248704466454944"
## [1] "residual for M2:  0.248847192837983"
```



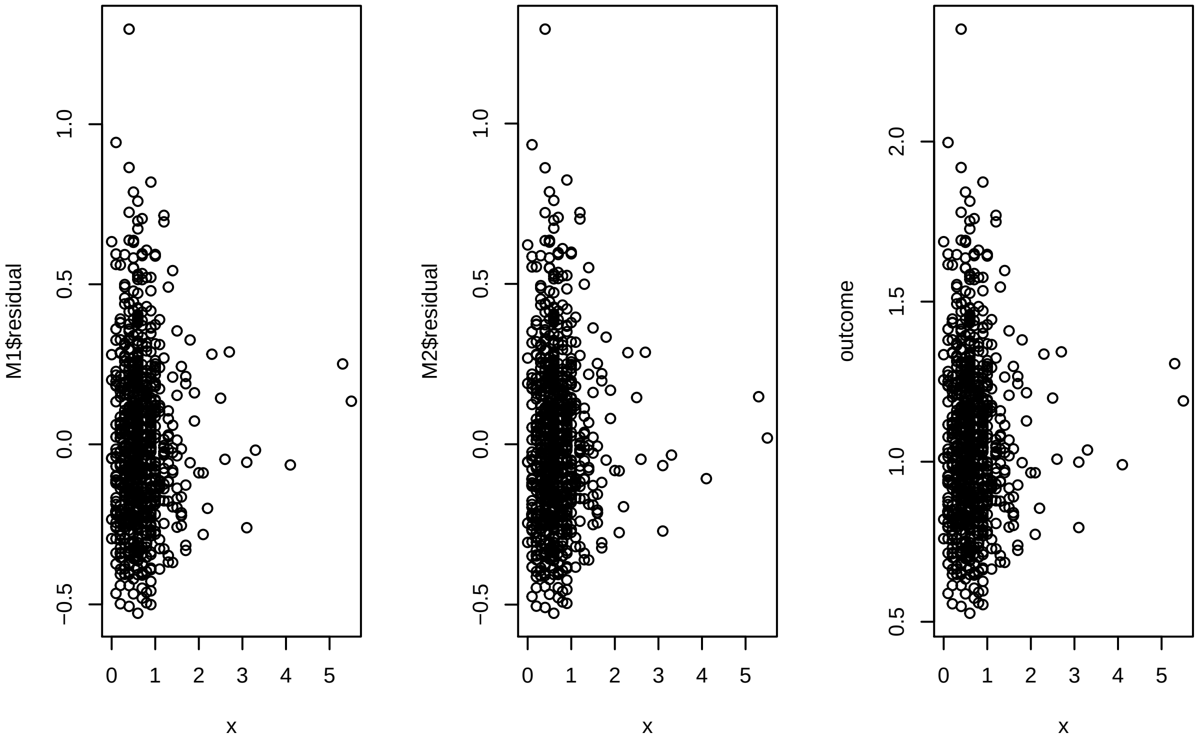
```
## [1] "residual for M1:  0.25026710930793"
## [1] "residual for M2:  0.250393729526099"
```



```
## [1] "residual for M1:  0.250043388210667"
## [1] "residual for M2:  0.25018695270193"
```



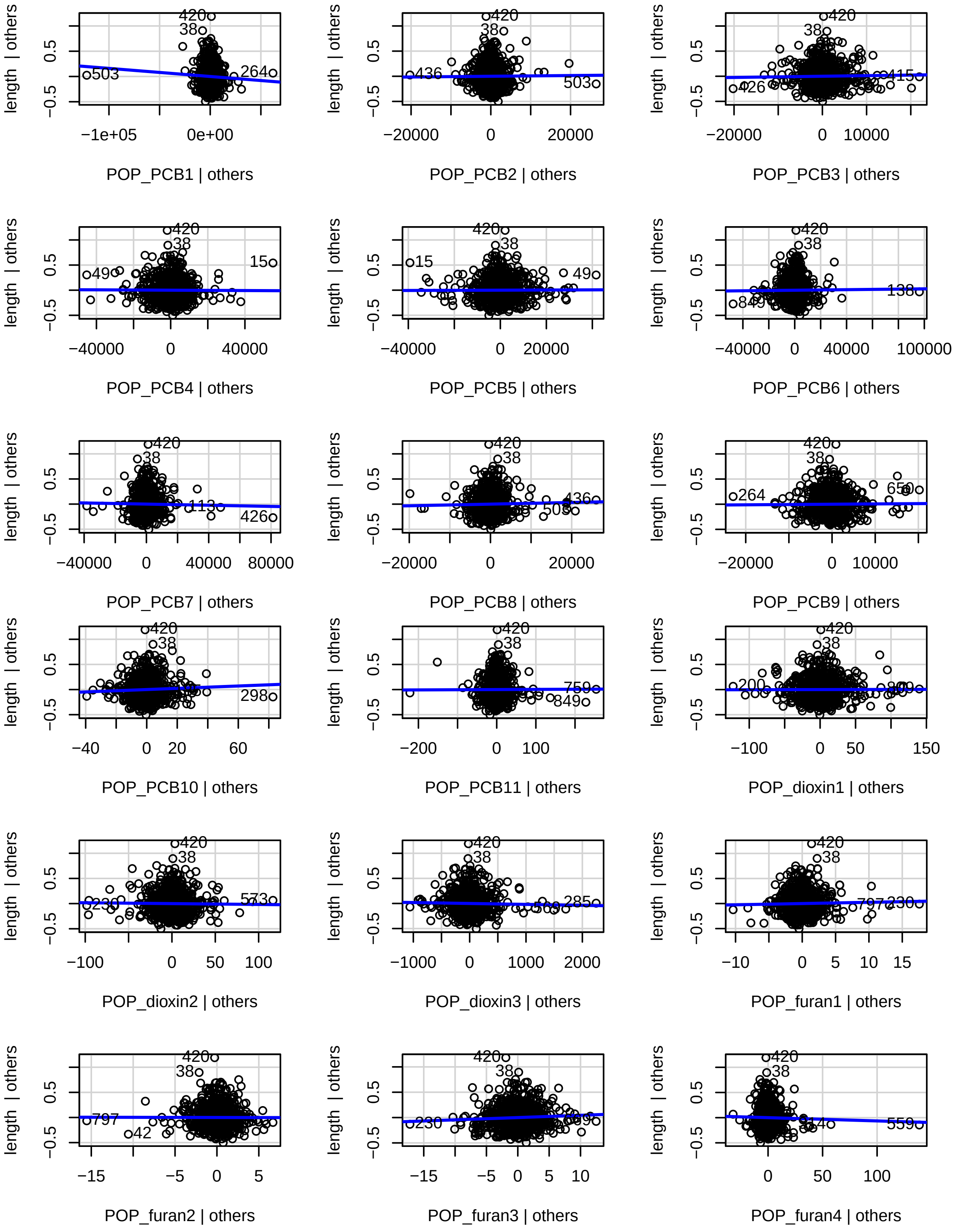
```
## [1] "residual for M1:  0.250382476371691"
## [1] "residual for M2:  0.25042580861039"
```

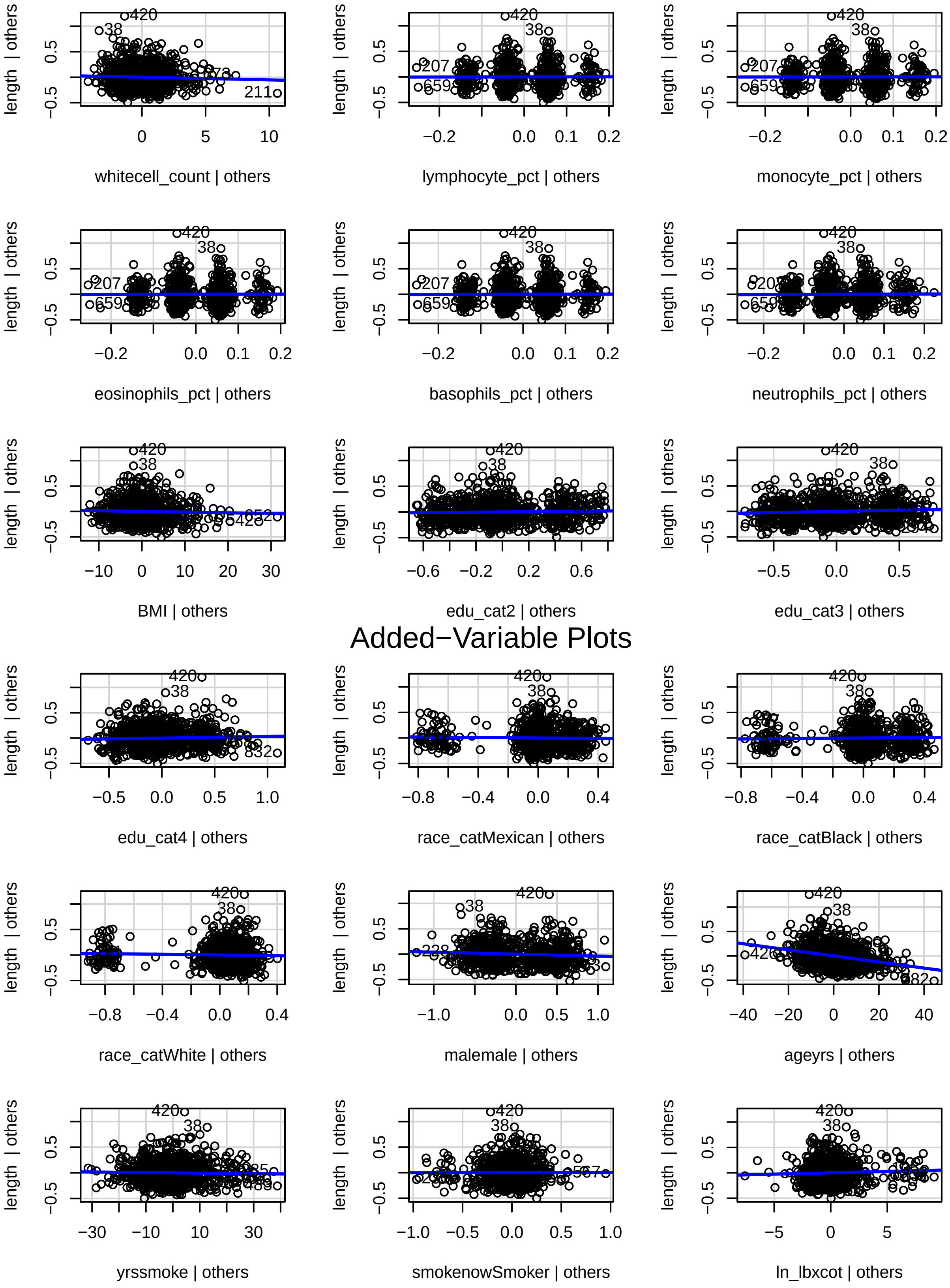


```
# Judy's work Part 2
# testing non-linearity in MLR
library(car)
```

```
## Loading required package: carData
```

```
M <- lm (length ~ ., data=pollutants)
avPlots(M)
```





```
# Estella's work 3
f <- as.formula(
  paste("length", paste("(", paste(POP_PCB, collapse = "+"), ")^2"), sep="~"))

m <- lm(f, data = pollutants)
summary(m)
```

```
##
## Call:
## lm(formula = f, data = pollutants)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53819 -0.16080 -0.01896  0.12149  1.20671
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.153e+00  2.892e-02  39.876  < 2e-16 ***
## POP_PCB1            -6.741e-06  3.521e-06  -1.915  0.05591 .
## POP_PCB2             3.801e-06  9.328e-06   0.407  0.68378
## POP_PCB3             6.747e-06  6.701e-06   1.007  0.31431
## POP_PCB4             1.373e-06  3.278e-06   0.419  0.67539
## POP_PCB5             1.920e-06  3.267e-06   0.588  0.55680
## POP_PCB6            -3.673e-06  4.336e-06  -0.847  0.39729
## POP_PCB7            -5.281e-06  4.697e-06  -1.124  0.26126
## POP_PCB8            -1.073e-05  8.331e-06  -1.288  0.19796
## POP_PCB9            -1.833e-06  5.806e-06  -0.316  0.75232
## POP_PCB10            2.720e-03  2.088e-03   1.303  0.19311
## POP_PCB11            4.644e-04  9.916e-04   0.468  0.63969
## POP_PCB1:POP_PCB2    9.529e-11  2.113e-10   0.451  0.65216
## POP_PCB1:POP_PCB3   -6.580e-10  4.156e-10  -1.583  0.11377
## POP_PCB1:POP_PCB4    1.116e-10  1.917e-10   0.582  0.56080
## POP_PCB1:POP_PCB5   -1.621e-11  1.318e-10  -0.123  0.90218
## POP_PCB1:POP_PCB6    6.244e-11  2.176e-10   0.287  0.77423
## POP_PCB1:POP_PCB7    2.221e-11  2.742e-10   0.081  0.93548
## POP_PCB1:POP_PCB8   -5.209e-10  2.693e-10  -1.935  0.05340 .
## POP_PCB1:POP_PCB9    4.146e-10  2.287e-10   1.813  0.07020 .
## POP_PCB1:POP_PCB10   1.675e-07  1.311e-07   1.277  0.20183
## POP_PCB1:POP_PCB11  -6.663e-08  7.321e-08  -0.910  0.36303
## POP_PCB2:POP_PCB3    1.673e-09  8.717e-10   1.919  0.05537 .
## POP_PCB2:POP_PCB4   -6.761e-10  4.688e-10  -1.442  0.14963
## POP_PCB2:POP_PCB5    3.840e-10  3.632e-10   1.057  0.29069
## POP_PCB2:POP_PCB6   -1.426e-09  5.834e-10  -2.444  0.01474 *
## POP_PCB2:POP_PCB7    1.532e-09  6.770e-10   2.264  0.02387 *
## POP_PCB2:POP_PCB8    2.135e-09  8.207e-10   2.602  0.00945 **
## POP_PCB2:POP_PCB9   -1.356e-09  7.249e-10  -1.870  0.06183 .
## POP_PCB2:POP_PCB10  -1.232e-06  4.242e-07  -2.904  0.00378 **
## POP_PCB2:POP_PCB11   3.388e-07  2.013e-07   1.683  0.09270 .
## POP_PCB3:POP_PCB4   -3.996e-11  1.199e-10  -0.333  0.73900
## POP_PCB3:POP_PCB5    4.665e-11  2.413e-10   0.193  0.84674
## POP_PCB3:POP_PCB6   -3.741e-10  2.662e-10  -1.405  0.16029
## POP_PCB3:POP_PCB7    6.438e-10  2.896e-10   2.223  0.02649 *
## POP_PCB3:POP_PCB8    7.340e-10  8.821e-10   0.832  0.40563
## POP_PCB3:POP_PCB9   -4.221e-10  5.470e-10  -0.772  0.44059
## POP_PCB3:POP_PCB10  -4.835e-07  2.555e-07  -1.892  0.05885 .
## POP_PCB3:POP_PCB11   7.155e-08  7.874e-08   0.909  0.36382
## POP_PCB4:POP_PCB5    3.002e-12  6.669e-11   0.045  0.96410
## POP_PCB4:POP_PCB6    1.788e-10  1.543e-10   1.159  0.24694
## POP_PCB4:POP_PCB7   -2.117e-10  1.579e-10  -1.341  0.18019
## POP_PCB4:POP_PCB8   -4.525e-11  3.961e-10  -0.114  0.90908
## POP_PCB4:POP_PCB9    1.217e-10  2.625e-10   0.464  0.64294
## POP_PCB4:POP_PCB10   1.345e-07  8.933e-08   1.505  0.13265
## POP_PCB4:POP_PCB11   1.685e-08  5.047e-08   0.334  0.73861
## POP_PCB5:POP_PCB6    4.714e-11  1.390e-10   0.339  0.73458
## POP_PCB5:POP_PCB7   -1.555e-10  1.446e-10  -1.076  0.28244
## POP_PCB5:POP_PCB8   -4.639e-10  3.185e-10  -1.457  0.14562
## POP_PCB5:POP_PCB9   -1.626e-11  1.822e-10  -0.089  0.92890
## POP_PCB5:POP_PCB10   9.703e-08  9.241e-08   1.050  0.29406
## POP_PCB5:POP_PCB11  -5.549e-08  4.079e-08  -1.360  0.17407
## POP_PCB6:POP_PCB7   -2.248e-11  1.147e-10  -0.196  0.84474
## POP_PCB6:POP_PCB8    7.086e-10  3.808e-10   1.861  0.06310 .
## POP_PCB6:POP_PCB9    4.295e-10  3.267e-10   1.315  0.18895
## POP_PCB6:POP_PCB10   2.152e-07  1.182e-07   1.820  0.06909 .
## POP_PCB6:POP_PCB11  -4.299e-08  2.038e-08  -2.109  0.03523 *
## POP_PCB7:POP_PCB8   -1.029e-09  4.279e-10  -2.404  0.01645 *
## POP_PCB7:POP_PCB9   -2.467e-10  3.622e-10  -0.681  0.49603
## POP_PCB7:POP_PCB10  -3.893e-08  1.308e-07  -0.298  0.76608
## POP_PCB7:POP_PCB11   4.226e-08  3.690e-08   1.145  0.25246
## POP_PCB8:POP_PCB9    1.317e-10  5.297e-10   0.249  0.80373
## POP_PCB8:POP_PCB10   5.264e-07  3.029e-07   1.738  0.08265 .
## POP_PCB8:POP_PCB11  -5.764e-08  1.285e-07  -0.449  0.65382
## POP_PCB9:POP_PCB10  -2.240e-08  1.448e-07  -0.155  0.87712
## POP_PCB9:POP_PCB11   7.916e-08  6.811e-08   1.162  0.24548
## POP_PCB10:POP_PCB11 -5.384e-05  2.694e-05  -1.999  0.04599 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2377 on 797 degrees of freedom
## Multiple R-squared:  0.1666, Adjusted R-squared:  0.09763
## F-statistic: 2.415 on 66 and 797 DF,  p-value: 1.316e-08
```

```
# Estella's work 4
# setting threshold of pvalue to be 0.05 and assess possible interaction terms
pvalues <- summary(m)$coefficients[,4]
p_threshold = 0.05
selected <-which(pvalues<= p_threshold)
names(selected)
```

```
## [1] "(Intercept)"         "POP_PCB2:POP_PCB6"   "POP_PCB2:POP_PCB7"
## [4] "POP_PCB2:POP_PCB8"   "POP_PCB2:POP_PCB10"  "POP_PCB3:POP_PCB7"
## [7] "POP_PCB6:POP_PCB11"  "POP_PCB7:POP_PCB8"   "POP_PCB10:POP_PCB11"
```

```
#stepwise parameters selection without any interaction terms
M0 <- lm(length ~ 1, data = train_data) # minimal model
Mfull <- lm(length ~ ., data= train_data)

## 2 corresponds to AIC
## log(n) corresponds to BIC

# stepwise AIC
Mstart <- lm(length ~ ., data= train_data)
```

```r
system.time({
  MAIC <- step(object = Mstart,
               scope = list(lower = M0, upper = Mfull),
               direction = "both", trace = 0, k = 2)
})
```

```
##    user  system elapsed
##   0.812   0.085   0.904
```

```r
#stepwiseBIC
system.time({
  MBIC <- step(object = Mstart,
               scope = list(lower = M0, upper = Mfull),
               direction = "both", trace = 0, k = log(nrow(train_data)))
})
```

```
##    user  system elapsed
##   0.826   0.078   0.912
```

```r
#stepwiseB_Adjusted R2
MAIC
```

```
##
## Call:
## lm(formula = length ~ POP_PCB1 + POP_PCB10 + POP_furan1 + POP_furan2 +
##     whitecell_count + monocyte_pct + edu_cat + race_cat + male +
##     ageyrs + ln_lbxcot, data = train_data)
##
## Coefficients:
##     (Intercept)         POP_PCB1        POP_PCB10       POP_furan1
##       1.443e+00       -5.602e-07        1.780e-03       -6.532e-03
##      POP_furan2  whitecell_count     monocyte_pct         edu_cat2
##       8.968e-03       -1.029e-02       -6.643e-03        4.105e-02
##        edu_cat3         edu_cat4  race_catMexican    race_catBlack
##       6.188e-02        8.254e-02       -3.635e-03        3.584e-02
##   race_catWhite         malemale           ageyrs        ln_lbxcot
##      -4.701e-02       -4.513e-02       -5.820e-03        7.573e-03
```

```r
MBIC
```

```
##
## Call:
## lm(formula = length ~ POP_furan3 + ageyrs, data = train_data)
##
## Coefficients:
## (Intercept)   POP_furan3       ageyrs
##    1.355743     0.005969    -0.006922
```

```r
# stepwise parameters selection with any interaction terms
M0 <- lm(length ~ 1, data = train_data) # minimal model

# tail to remove length column
single <- paste(tail(colnames(train_data),-1), collapse = " + ")
# tail to remove intercept column
interaction <- paste(tail(names(selected),-1), collapse = " + ")
f_interaction <- as.formula(
  paste("length", paste("(", single,"+", interaction, ")"), sep = " ~"))
```

```
Mfull <- lm(f_interaction, data = train_data)
Mstart <- lm(f_interaction, data = train_data)

# stepwise AIC
Mstart <- lm(length ~ ., data= train_data)
system.time({
  MAIC_Interaction <- step(object = Mstart,
                           scope = list(lower = M0, upper = Mfull),
                           direction = "both", trace = 0, k = 2)
})
```

```
##    user  system elapsed
##   0.833   0.079   0.915
```

```
#stepwiseBIC
system.time({
  MBIC_Interaction <- step(object = Mstart,
                           scope = list(lower = M0, upper = Mfull),
                           direction = "both", trace = 0,
                           k = log(nrow(train_data)))
})
```

```
##    user  system elapsed
##   0.875   0.080   0.956
```

```
#stepwiseB_Adjusted R2
MAIC_Interaction
```

```
##
## Call:
## lm(formula = length ~ POP_PCB1 + POP_PCB6 + POP_PCB10 + POP_PCB11 +
##     POP_dioxin2 + POP_furan3 + whitecell_count + monocyte_pct +
##     BMI + edu_cat + race_cat + male + ageyrs + ln_lbxcot + POP_PCB10:POP_PCB11,
##     data = train_data)
##
## Coefficients:
##         (Intercept)             POP_PCB1             POP_PCB6
##           1.473e+00           -8.511e-07            1.150e-06
##           POP_PCB10            POP_PCB11          POP_dioxin2
##           2.839e-03            9.157e-04           -6.180e-04
##          POP_furan3      whitecell_count         monocyte_pct
##           4.745e-03           -9.472e-03           -6.707e-03
##                 BMI             edu_cat2             edu_cat3
##          -2.272e-03            4.205e-02            5.902e-02
##            edu_cat4     race_catMexican        race_catBlack
##           7.656e-02            1.408e-03            4.927e-02
##       race_catWhite             malemale               ageyrs
##          -3.842e-02           -3.208e-02           -6.126e-03
##           ln_lbxcot  POP_PCB10:POP_PCB11
##           7.374e-03           -2.457e-05
```

```
MBIC_Interaction
```

```
##
## Call:
## lm(formula = length ~ POP_furan3 + ageyrs, data = train_data)
```

```
##
## Coefficients:
## (Intercept)   POP_furan3       ageyrs
##    1.355743     0.005969    -0.006922
```

```
# mxn's work
predAIC <- predict(MAIC, newdata=test_data)
predBIC <- predict(MBIC, newdata=test_data)
predAICInteraction <- predict(MAIC_Interaction, newdata=test_data)
predBICInteraction <- predict(MBIC_Interaction, newdata=test_data)

mean((test_data$length - predAIC)^2)
```

```
## [1] 0.05336494
```

```
mean((test_data$length - predBIC)^2)
```

```
## [1] 0.04804827
```

```
mean((test_data$length - predAICInteraction)^2)
```

```
## [1] 0.05230268
```

```
mean((test_data$length - predBICInteraction)^2)
```

```
## [1] 0.04804827
```

5. Results:

Report on the findings of your analysis

6. Discussion:

Comment on your findings/conclusions; describe any limitations of your analysis.

7. Appendix

Data Summary

Looking at the useful metrics for the data

```
summary(pollutants)
```

```
##      length          POP_PCB1         POP_PCB2         POP_PCB3     
##  Min.   :0.5266   Min.   :  2000   Min.   :  2000   Min.   :  2000  
##  1st Qu.:0.8754   1st Qu.:  9975   1st Qu.:  4800   1st Qu.:  3700  
##  Median :1.0286   Median : 27600   Median : 11500   Median :  6200  
##  Mean   :1.0543   Mean   : 38082   Mean   : 15637   Mean   : 10158  
##  3rd Qu.:1.2095   3rd Qu.: 53325   3rd Qu.: 21825   3rd Qu.: 12000  
##  Max.   :2.3512   Max.   :572000   Max.   :165000   Max.   :123000  
##     POP_PCB4         POP_PCB5         POP_PCB6         POP_PCB7     
##  Min.   :  2100   Min.   :  2100   Min.   :  2000   Min.   :  1100  
##  1st Qu.: 11475   1st Qu.: 15600   1st Qu.:  4400   1st Qu.:  4000  
##  Median : 25550   Median : 36300   Median :  9400   Median :  7450  
##  Mean   : 38456   Mean   : 52650   Mean   : 16820   Mean   : 12682  
##  3rd Qu.: 50650   3rd Qu.: 68625   3rd Qu.: 19500   3rd Qu.: 15625  
##  Max.   :487000   Max.   :708000   Max.   :319000   Max.   :144000
##     POP_PCB8         POP_PCB9         POP_PCB10        POP_PCB11
##  Min.   :  1100   Min.   :  1100   Min.   :  1.70   Min.   :  1.30
##  1st Qu.:  3800   1st Qu.:  3900   1st Qu.:  9.10   1st Qu.: 14.80
##  Median :  6950   Median :  8050   Median : 18.35   Median : 24.50
##  Mean   : 10530   Mean   : 12220   Mean   : 24.49   Mean   : 38.15
##  3rd Qu.: 14425   3rd Qu.: 16025   3rd Qu.: 34.90   3rd Qu.: 42.95
##  Max.   :187000   Max.   :144000   Max.   :172.00   Max.   :845.00
##   POP_dioxin1      POP_dioxin2      POP_dioxin3       POP_furan1
##  Min.   :  1.90   Min.   :  1.40   Min.   :  36.8   Min.   : 1.000
##  1st Qu.: 23.90   1st Qu.: 21.27   1st Qu.: 197.0   1st Qu.: 3.200
##  Median : 41.35   Median : 37.80   Median : 342.5   Median : 5.200
##  Mean   : 57.65   Mean   : 47.81   Mean   : 494.4   Mean   : 6.371
##  3rd Qu.: 71.62   3rd Qu.: 62.42   3rd Qu.: 603.0   3rd Qu.: 7.700
##  Max.   :760.00   Max.   :281.00   Max.   :8190.0   Max.   :44.400
##    POP_furan2       POP_furan3       POP_furan4     whitecell_count
##  Min.   : 0.800   Min.   : 0.700   Min.   :  0.90   Min.   : 2.300
##  1st Qu.: 2.600   1st Qu.: 2.200   1st Qu.:  6.40   1st Qu.: 5.600
##  Median : 4.200   Median : 5.050   Median :  9.65   Median : 6.900
##  Mean   : 5.390   Mean   : 6.669   Mean   : 11.54   Mean   : 7.191
##  3rd Qu.: 6.825   3rd Qu.: 9.300   3rd Qu.: 14.00   3rd Qu.: 8.300
##  Max.   :33.500   Max.   :38.300   Max.   :234.00   Max.   :20.100
##  lymphocyte_pct   monocyte_pct     eosinophils_pct basophils_pct
##  Min.   : 5.80   Min.   : 1.600   Min.   :21.60   Min.   : 0.000
##  1st Qu.:24.00   1st Qu.: 6.600   1st Qu.:52.35   1st Qu.: 1.500
##  Median :28.95   Median : 7.700   Median :59.30   Median : 2.300
##  Mean   :29.92   Mean   : 7.936   Mean   :58.62   Mean   : 2.903
##  3rd Qu.:35.42   3rd Qu.: 9.100   3rd Qu.:65.22   3rd Qu.: 3.700
##  Max.   :73.40   Max.   :23.800   Max.   :88.10   Max.   :28.200
##  neutrophils_pct       BMI        edu_cat    race_cat        male
##  Min.   :0.0000   Min.   :16.16   1:270   Other  : 71   female:490
##  1st Qu.:0.4000   1st Qu.:23.88   2:199   Mexican:191   male  :374
##  Median :0.6000   Median :27.38   3:228   Black  :154
##  Mean   :0.6669   Mean   :28.09   4:167   White  :448
##  3rd Qu.:0.8000   3rd Qu.:31.17
##  Max.   :5.5000   Max.   :62.99
##      ageyrs         yrssmoke          smokenow     ln_lbxcot
##  Min.   :20.00   Min.   : 0.0   Non-Smoker:664   Min.   :-4.5099
##  1st Qu.:34.00   1st Qu.: 0.0   Smoker    :200   1st Qu.:-4.0745
##  Median :46.00   Median : 0.0                    Median :-2.7334
##  Mean   :48.36   Mean   :10.6                    Mean   :-0.9804
##  3rd Qu.:63.00   3rd Qu.:20.0                    3rd Qu.: 2.8000
##  Max.   :85.00   Max.   :69.0                    Max.   : 6.5848
```

Outlier Entries

Here we will find entries where outliers for different covariate occurred.

```r
pollutant_mat = data.matrix(pollutants, rownames.force = NA)

max_PCB_idx = c()
for (c in 2:12) {
  max_PCB_idx[c-1] = which.max(pollutant_mat[, c])
}
```

```
max_PCB_idx
```

```
##  [1] 436 436 436 436 436 436 426 436 436 298 272
```

```
max_dioxin_idx = c()
for (c in 13:15) {
  max_dioxin_idx[c-12] = which.max(pollutant_mat[, c])
}
max_dioxin_idx
```

```
## [1] 285 573 285
```

```
max_furan_idx = c()
for (c in 16:19) {
  max_furan_idx[c-15] = which.max(pollutant_mat[, c])
}
max_furan_idx
```

```
## [1] 230 559 590 559
```

```
max_WBC_idx = c()
for (c in 20:25) {
  max_WBC_idx[c-19] = which.max(pollutant_mat[, c])
}
max_WBC_idx
```

```
## [1] 211 766 440 782 739 415
```