STAT 331 Final Project

Maxine, Estella, Judy, Weiwei

04/12/2021

Contents

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

We are looking to investigate the most influential factors that contribute to the average leukocyte telomere length in a person. We would like to especially look for human-adjustable factors such as whether a person smokes or exposure to persistent organic pollutants.

3 Exploratory Data Analysis

The covariates of interest from the provided dataset are

```
names(pollutants)
```

```
##  [1] "length"          "POP_PCB1"        "POP_PCB2"
##  [4] "POP_PCB3"        "POP_PCB4"        "POP_PCB5"
##  [7] "POP_PCB6"        "POP_PCB7"        "POP_PCB8"
## [10] "POP_PCB9"        "POP_PCB10"       "POP_PCB11"
## [13] "POP_dioxin1"     "POP_dioxin2"     "POP_dioxin3"
## [16] "POP_furan1"      "POP_furan2"      "POP_furan3"
## [19] "POP_furan4"      "whitecell_count" "lymphocyte_pct"
## [22] "monocyte_pct"    "eosinophils_pct" "basophils_pct"
## [25] "neutrophils_pct" "BMI"             "edu_cat"
## [28] "race_cat"        "male"            "ageyrs"
## [31] "yrssmoke"        "smokenow"        "ln_lbxcot"
```

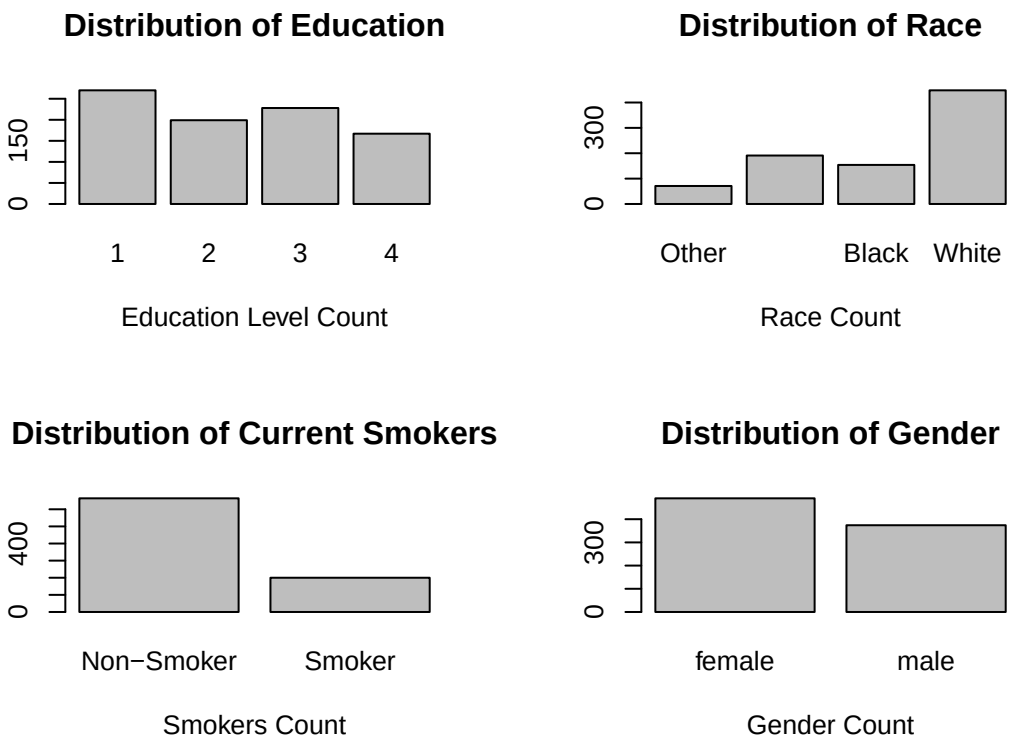
Note that “edu_cat”, “race_cat”, “male”, “smokenow” are categorical values and the rest are continuous.

3.1 Data Distribution

We shall now investigate the distribution of covariates from the supplied data.

From the output of summary statistics on the covariates (see in appendix 7.1), we observed that all values are non-negative and there are more observations with values close to 0 than values with large magnitude across all covariates.

Now we shall have a closer look at the distribution of individual covariate. For categorical data,



We may observe from the bar graphs that there are 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 is relatively close.

Now for continuous data, we made boxplots to see the distribution of these covariates, the plots can be found in the appendix 7.2. From these plots, we notice some extreme outliers in some concentration values of PCBs, Dioxins, and Furan. The maximum values are sometimes over double the magnitude of the second largest.

However, with a little investigation in the appendix 7.3, we see that the extreme outliers across different types of PCB mostly came from one observation.

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

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

This observation contributes to the maximum value for PCB1 to PCB6, as well as PCB8 and PCB9

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

- Entry 285 contain the highest value for Dioxin 1 and 3, which are the two 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, as we see from the boxplots, do not have outliers that are as extreme as those from pollutant data. We further observe that they do not have a common entry that contributes to the outliers.

3.2 Multicolinearity

We learned that severe multicollinearity between covariates could result in unstable coefficient estimates and inflated standard errors. Therefore, in this section, we will investigate correlations among values that we may

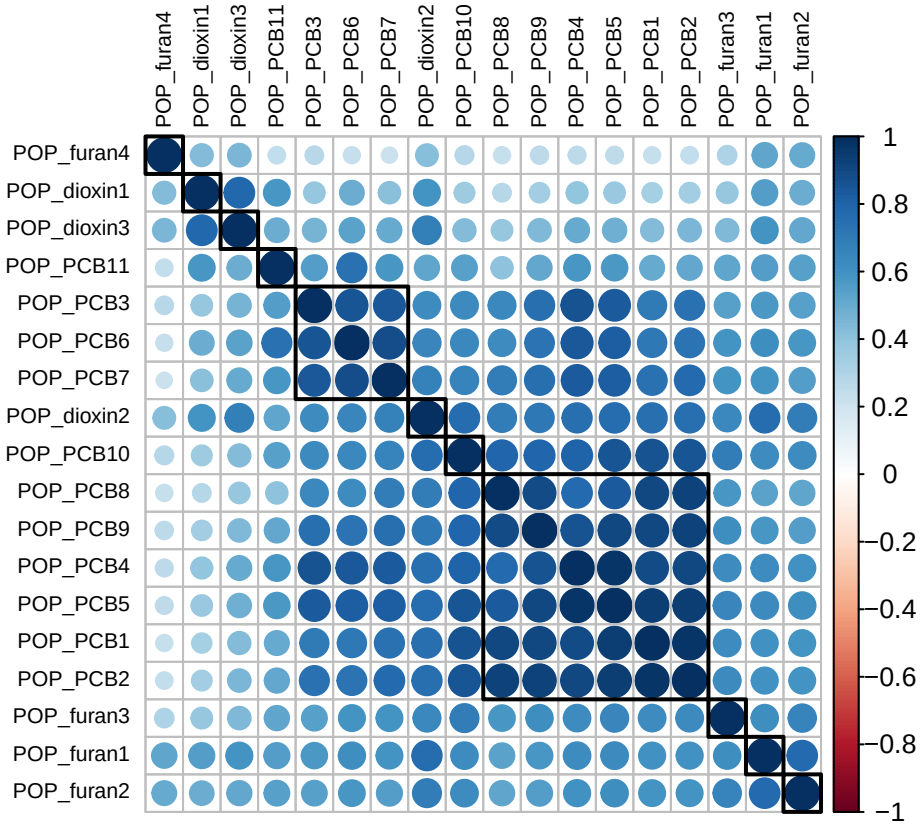
expect multicollinearity to appear, such as between different types of organic pollutants POP_PCB1–11, POP_dioxin1-3, Pop_furan1–4, as well as white blood cell components.

To obtain the heatmaps that visualize correlations among covariates, we first computed Spearman correlations for each pair of covariates of interest and represented the measured values through gradients of a color scheme. In our example, blue refers to positive correlations and red, negative. Furthermore, the darker colours signify a higher correlation among the covariates. Finally, we clustered variables with higher correlations together such that the covariates within the same rectangles are highly correlated such that they may have dependencies on each other.

3.2.1 Correlation among Persistent Pollution

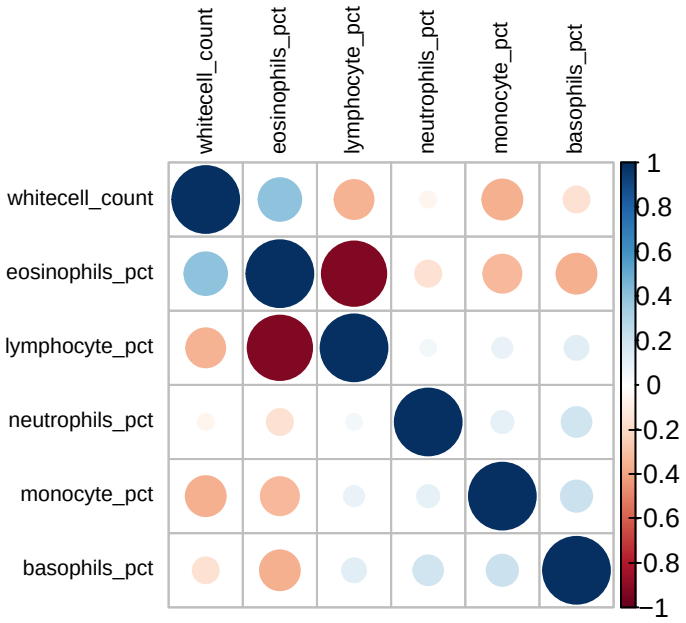
```
## corrplot 0.84 loaded
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```



Based on the above plot, we noticed the correlations mostly exist among the organic pollutants of the same kind. Specifically, the correlations among POP_PCB3,6,7 and POP_PCB8,9,4,5,1,2 are higher than others.

3.2.2 Correlation between White Blood Cells



From the graph above, we see that there is no strong positive correlation among the components of white blood cells, however, there is a strong negative correlation between lymphocytes and eosinophils percentage in the given data.

We shall omit the analysis on correlations between other covariate from this section as we do not expect personal health data such as BMI or years of smoke to have a logical significant correlation with each other, white blood cell data, or exposure to pollutants.

To further investigate how these listed correlations affect the observed data, we shall consider adding interaction terms to our model, and performed p-test to check their statistical significance.

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.

4.1 Linear Model Assumptions

Since we have no access to data collection, we shall proceed by assuming that the independence assumption is satisfied.

As for the normality assumption, as the given dataset is relatively large, we may assume the data is approximately Normally distributed due to the Central Limit Theorem.

Now to assess whether any covariate has a nonlinearity relationship with the outcome in the multiple linear regression model, we used added-variable plots(avPlot), as shown in appendix 7.4. The plots isolate the relationship between the outcome and each of the covariates after adjusting for the other covariate. If the plot of the outcome versus a covariate x has a nonlinear shape, it may indicate a regression model with a higher power of this variable, for example, x^2. With the given data, we see from the avPlots that all plots have a linear shape, thus the outcome is expected to have a linear relationship with all of the covariates. Therefore, the models constructed in this report do not consider non-linear terms.

4.2 Heteroskedasticity

We also need to verify the equal variance(heteroscedasticity) assumption. As shown in the appendix 7.5, if there are evident patterns in the residuals, we might not be able to simply trust the results. Fortunately, we can see that the random residuals are uncorrelated and uniform.

4.3 Finding the model

We shall first split the data into training and testing set to ensure the final model is well-generalized without problems such as overfitting or underfitting.

```
set.seed(23)
train_idx <- sample(nrow(pollutants), 650, replace = FALSE, prob = NULL)
train_data <- pollutants[train_idx,]
test_data <- pollutants[-train_idx, ]
```

4.3.1 Investigate Interactions

As we have seen in the EDA section, we would like to investigate interactions among pollutants as well as white blood cell-related data. By building a large linear model and filtering the interactions with p values ≤ 0.05, we have selected the following potential interaction terms that we may consider in the model building process:

```
names(selected)
```

```
##  [1] "POP_PCB1:POP_PCB9"              "POP_PCB2:POP_PCB4"
##  [3] "POP_PCB2:POP_PCB5"              "POP_PCB2:POP_PCB6"
##  [5] "POP_PCB2:POP_PCB8"              "POP_PCB2:POP_PCB9"
##  [7] "POP_PCB2:POP_PCB10"             "POP_PCB2:POP_furan3"
##  [9] "POP_PCB2:POP_furan4"            "POP_PCB2:lymphocyte_pct"
## [11] "POP_PCB2:monocyte_pct"          "POP_PCB2:eosinophils_pct"
## [13] "POP_PCB2:basophils_pct"         "POP_PCB4:POP_PCB10"
## [15] "POP_PCB4:POP_dioxin3"           "POP_PCB5:POP_PCB11"
## [17] "POP_PCB5:POP_dioxin2"           "POP_PCB5:POP_dioxin3"
## [19] "POP_PCB5:POP_furan2"            "POP_PCB6:POP_PCB8"
## [21] "POP_PCB6:POP_PCB10"             "POP_PCB7:POP_PCB9"
## [23] "POP_PCB7:POP_dioxin2"           "POP_PCB8:POP_PCB10"
## [25] "POP_PCB8:POP_PCB11"             "POP_PCB8:POP_furan3"
## [27] "POP_PCB9:POP_dioxin2"           "whitecell_count:lymphocyte_pct"
## [29] "whitecell_count:monocyte_pct"   "whitecell_count:eosinophils_pct"
## [31] "whitecell_count:basophils_pct"
```

We now shall select a linear model with all covariate and interaction terms, we can find the summary of the resulting model in the appendix 7.7.1.

```
MAIC_Interaction #model 1
```

```
##
## Call:
## lm(formula = length ~ POP_PCB1 + POP_PCB10 + POP_furan3 + whitecell_count +
##     eosinophils_pct + race_cat + male + ageyrs + ln_lbxcot, data = train_data)
##
## Coefficients:
##     (Intercept)         POP_PCB1        POP_PCB10       POP_furan3
##       1.305e+00       -7.505e-07        1.527e-03        3.658e-03
## whitecell_count  eosinophils_pct  race_catMexican    race_catBlack
##      -6.718e-03        2.110e-03       -1.834e-02        5.185e-02
```

```
##   race_catWhite         malemale           ageyrs        ln_lbxcot
##      -1.286e-02       -5.164e-02       -6.727e-03        5.046e-03
```

```
AIC_MSPE
```

```
## [1] 0.0471547
```

```
MBIC_Interaction
```

```
##
## Call:
## lm(formula = length ~ POP_PCB10 + male + ageyrs, data = train_data)
##
## Coefficients:
## (Intercept)    POP_PCB10     malemale       ageyrs
##    1.399288     0.001788    -0.053197    -0.007457
```

```
BIC_MSPE
```

```
## [1] 0.04679024
```

This result shows that the model selected by BIC was preferred as it has a lower MSPE, very generalized, and easy to interpret. At the same time, note that the model chosen by AIC has more parameters but a lower prediction score, this implies that the added parameters added too much variability to the model and seems to have overfitted the training data. Therefore, we decided to use the parameters picked by BIC for our first candidate model, named model 1. The formula of model 1 is:

```
model1_f <- formula(MBIC_Interaction)
model1_f
```

```
## length ~ POP_PCB10 + male + ageyrs
```

Furthermore, as we had included only one interaction term in the AIC model and it did not improve the performance of the model. We decided that none of the interaction terms contribute significantly to the outcome of interest (telomere length). In the next part of the analysis, we have removed these terms for simplicity.

4.3.2 Reduce Multicolinearity

An additional technique we may use to reduce the impact of multicollinearity on our model is checking variance inflation factor (VIF). As interaction terms were eliminated, we shall regress on all non-categorical covariates and identify those with the largest VIF one at a time until there were no more with 'high' multicollinearity. We used a VIF (Variance Inflation Factor) > 10 as an indicator of "high" multicollinearity (general practice). And after the covariate eliminations, The explanatory variables that remained from the selection are:

```
VIFselected
```

```
##  [1] "POP_PCB3"         "POP_PCB6"         "POP_PCB7"
##  [4] "POP_PCB8"         "POP_PCB9"         "POP_PCB10"
##  [7] "POP_PCB11"        "POP_dioxin1"      "POP_dioxin2"
## [10] "POP_dioxin3"      "POP_furan1"       "POP_furan2"
## [13] "POP_furan3"       "POP_furan4"       "whitecell_count"
## [16] "lymphocyte_pct"   "monocyte_pct"     "basophils_pct"
## [19] "neutrophils_pct" "BMI"              "edu_cat"
## [22] "race_cat"         "male"             "ageyrs"
## [25] "yrssmoke"         "smokenow"         "ln_lbxcot"
```

To validate our parameter selection steps, we could run stepwise selection again on the reduced model.

4.3.3 Model via Forward-Backward Selection

```
MAIC_reduced
```

```
##
## Call:
## lm(formula = length ~ POP_dioxin3 + POP_furan3 + lymphocyte_pct +
##     race_cat + male + ageyrs + ln_lbxcot, data = train_data)
##
## Coefficients:
##     (Intercept)      POP_dioxin3       POP_furan3   lymphocyte_pct
##       1.436e+00       -3.528e-05        5.877e-03       -1.801e-03
## race_catMexican    race_catBlack    race_catWhite         malemale
##      -1.633e-02        5.850e-02       -1.014e-02       -5.309e-02
##          ageyrs        ln_lbxcot
##      -6.600e-03        3.965e-03
```

```
AIC_MSPE
```

```
## [1] 0.04709662
```

```
MBIC_reduced
```

```
##
## Call:
## lm(formula = length ~ POP_furan3 + ageyrs, data = train_data)
##
## Coefficients:
## (Intercept)   POP_furan3       ageyrs
##    1.373603     0.005311    -0.007226
```

```
BIC_MSPE
```

```
## [1] 0.04554553
```

We got a similar result that the model selected by BIC still outperformed the one selected by AIC. However, this time the stepwise function which ran on the reduced model with BIC had selected different covariates for us. We could build another model, model 2 upon those newly selected covariates. The formula of model 2 is:

```
model2_f <-  formula(MBIC_reduced)
model2_f
```

```
## length ~ POP_furan3 + ageyrs
```

4.3.4 Model via Cross-Validation with Ridge

In order to get accurate prediction evaluations for our models (model 1&2), we used the idea of 80% and 20% train-test split; To ensure the entire training set was covered and each observation was well represented, we divided the training data into 10 folds and repeatedly cross-validated the MSPE. Besides, we performed shrinkage methods like lasso and ridge to solve the overfitting problem. For example, we used ridge with cross validation to update our Model 1&2 as follow:

```
# estella's work cross validation using ridge on BIC model

library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.6.2
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-1
```

```
## model 1
Y <-  train_data[, c("length")]
train_model1_X <-  model.matrix(lm(model1_f, data= train_data))
test_model1_X <- model.matrix(lm(model1_f, data= test_data))

 # use ridge, default 10 folds
cv_ridge_model1 <- cv.glmnet(x = data.matrix(train_model1_X), y = Y, alpha = 0)

paste("model 1")
```

```
## [1] "model 1"
```

```
# estimated betas for min lambda
coef(cv_ridge_model1, s = "lambda.min")
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  1.378410131
## (Intercept)  .
## POP_PCB10    0.001323664
## malemale    -0.049830828
## ageyrs      -0.006826190
```

```
pred_model1 <- predict(cv_ridge_model1, newx = data.matrix(test_model1_X ), s = "lambda.min")

## model 2

Y <-  train_data[, c("length")]
train_model2_X <-  model.matrix(lm(model2_f, data= train_data))
test_model2_X <- model.matrix(lm(model2_f, data= test_data))

 # use ridge, default 10 folds
cv_ridge_model2 <- cv.glmnet(x = data.matrix(train_model2_X), y = Y, alpha = 0)

paste("model 2")
```

```
## [1] "model 2"
```

```
# estimated betas for min lambda
coef(cv_ridge_model2, s = "lambda.min")
```

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  1.356876278
## (Intercept)  .
## POP_furan3   0.004246086
## ageyrs      -0.006737178
```

```
pred_model2 <- predict(cv_ridge_model2, newx = data.matrix(test_model2_X ), s = "lambda.min")
```

With the consideration that lasso could also do parameter selections, we examined sending the remaining covariates in the VIF reduced model alongwith the categorical covariates to 'glmnet' function and let it pick the best model for us. We named it model 3.

4.3.5 Model via Cross-Validation with LASSO

```
## estella's work cross validation using lasso to do model selections on reduced model selected by VIF
## model 3
# Load libraries
library(data.table)
library(mltools)

train_df <- as.data.table(train_data[c(VIFselected)])
train_oh <- one_hot(train_df )
test_df <- as.data.table(test_data[c(VIFselected)])
test_oh <- one_hot(test_df )

# try lasso and let lasso do the parameters selection
cvfit_lasso_oh <- cv.glmnet(x = data.matrix(train_oh), y = Y, alpha = 1) # use lasso
coef(cvfit_lasso_oh, s = "lambda.min")
```

```
## 36 x 1 sparse Matrix of class "dgCMatrix"
##                                  1
## (Intercept)          1.343495e+00
## POP_PCB3             .
## POP_PCB6             .
## POP_PCB7             .
## POP_PCB8             .
## POP_PCB9             .
## POP_PCB10            .
## POP_PCB11            .
## POP_dioxin1          .
## POP_dioxin2          .
## POP_dioxin3          .
## POP_furan1           .
## POP_furan2           .
## POP_furan3           1.822255e-03
## POP_furan4           .
## whitecell_count      .
## lymphocyte_pct      -4.627083e-04
## monocyte_pct        -1.557694e-03
## basophils_pct        .
## neutrophils_pct      .
## BMI                  .
## edu_cat_1           -2.542837e-02
## edu_cat_2            .
## edu_cat_3            .
## edu_cat_4            .
## race_cat_Other       .
## race_cat_Mexican     .
## race_cat_Black       4.315289e-02
## race_cat_White       .
## male_female          3.104109e-02
## male_male           -1.707626e-14
## ageyrs              -5.904777e-03
## yrssmoke             .
## smokenow_Non-Smoker  .
## smokenow_Smoker      .
```

```
## ln_lbxcot            2.427779e-03
```

```
pred_lasso <- predict(cvfit_lasso_oh, newx = data.matrix(test_oh ), s = "lambda.min")
```

5 Results

Report on the findings of your analysis

In the end, we looked at the model performance on the remaining test set and computed the MPSE of each model.

```
#model 1
model1_f
```

```
## length ~ POP_PCB10 + male + ageyrs
```

```
mean((test_data$length - pred_model1)^2)
```

```
## [1] 0.04661126
```

```
#model 2
model2_f
```

```
## length ~ POP_furan3 + ageyrs
```

```
mean((test_data$length - pred_model2 )^2)
```

```
## [1] 0.04568024
```

```
#model 3
paste("lasso selected model:")
```

```
## [1] "lasso selected model:"
```

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

```
## [1] 0.04653322
```

6 Discussion

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

We have considered the multicollinearity and interactions within the eleven PCB covariates and similarly for the three dioxin covariates and four furan covariates. However, the multicollinearity and interactions between these eighteen exposure covariates and other covariates are not considered. It is expected that there does not exist any causal relationship between exposure covariates and other covariates since the former relates to the surrounding environment and the latter relates to personal characteristics. For example, it's believed that the concentration of POP_PCB10 is unrelated to the value of ageyrs and BMI.

Besides, a linear regression model has four assumptions, namely linearity, normality, heteroskedasticity and independence. We have analyzed and confirmed that the first three assumptions hold. Without time-series data, it is difficult to visualize and assess independence. However, with a large sample, residuals are approximately independent, and we can assume independence.

7 Appendix

7.1 Data Summary

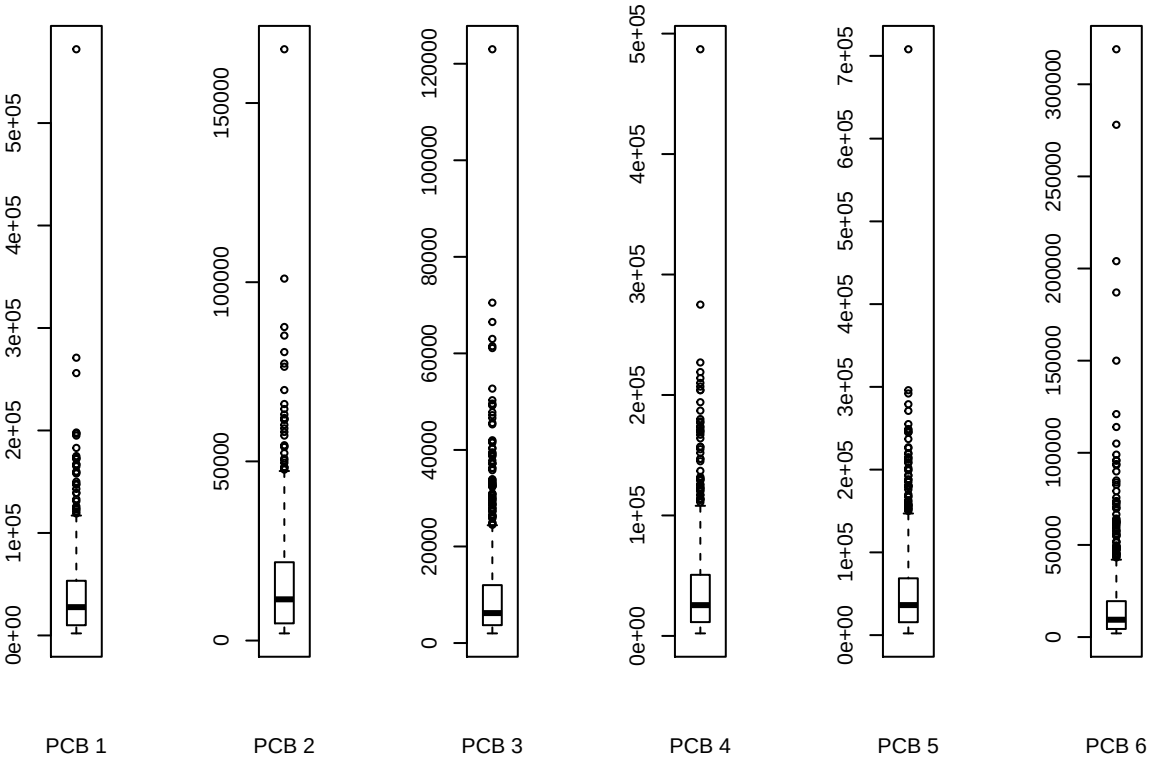
Looking at the useful metrics for the data

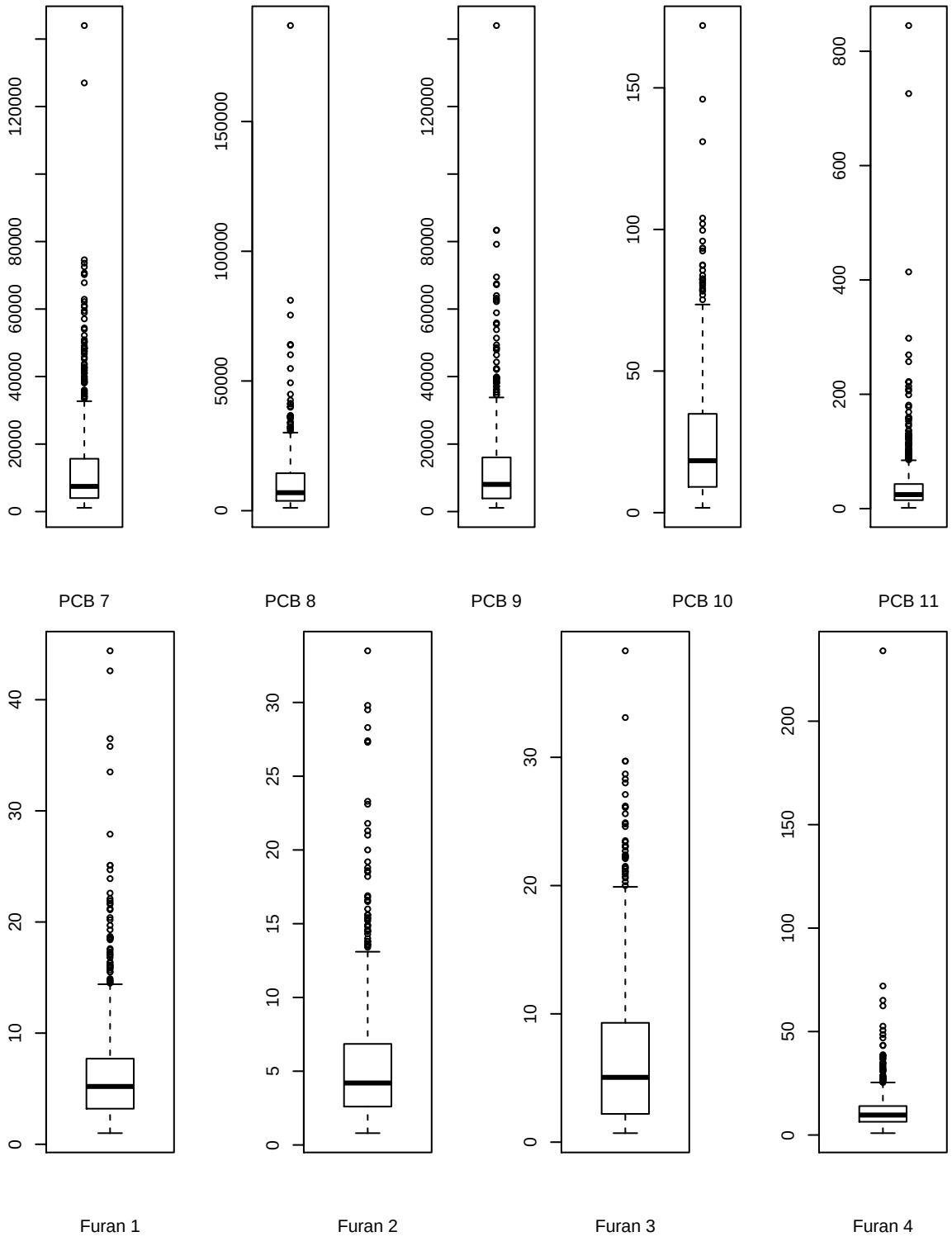
```
summary(pollutants)
```

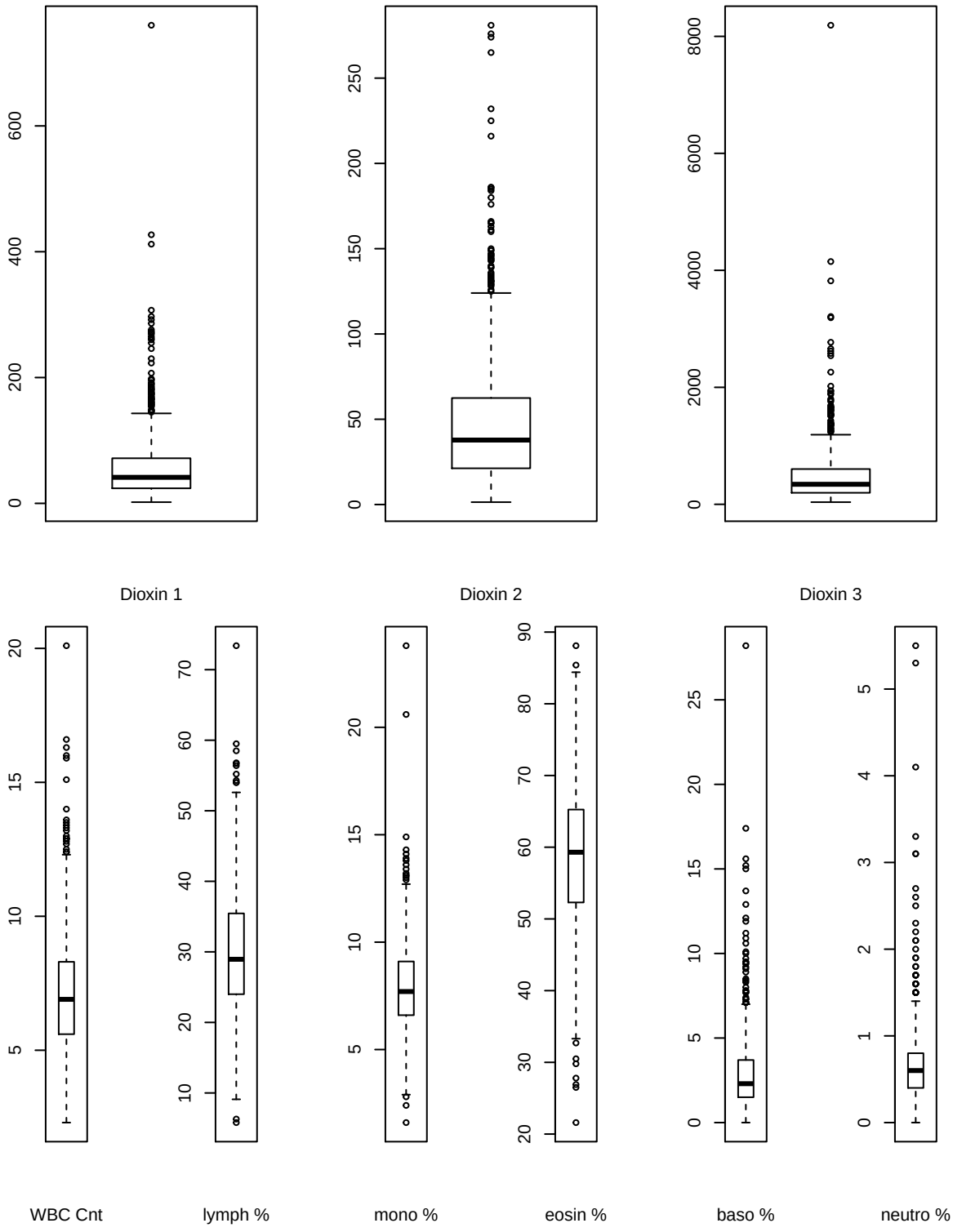
```
##      length          POP_PCB1         POP_PCB2         POP_PCB3
##  Min.   :0.5266   Min.   :  2000   Min.   :  2000   Min.   :  2000
##  1st Qu.:0.8754   1st Qu.:  9975   1st Qu.:  4800   1st Qu.:  3700
##  Median :1.0286   Median : 27600   Median : 11500   Median :  6200
##  Mean   :1.0543   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
```

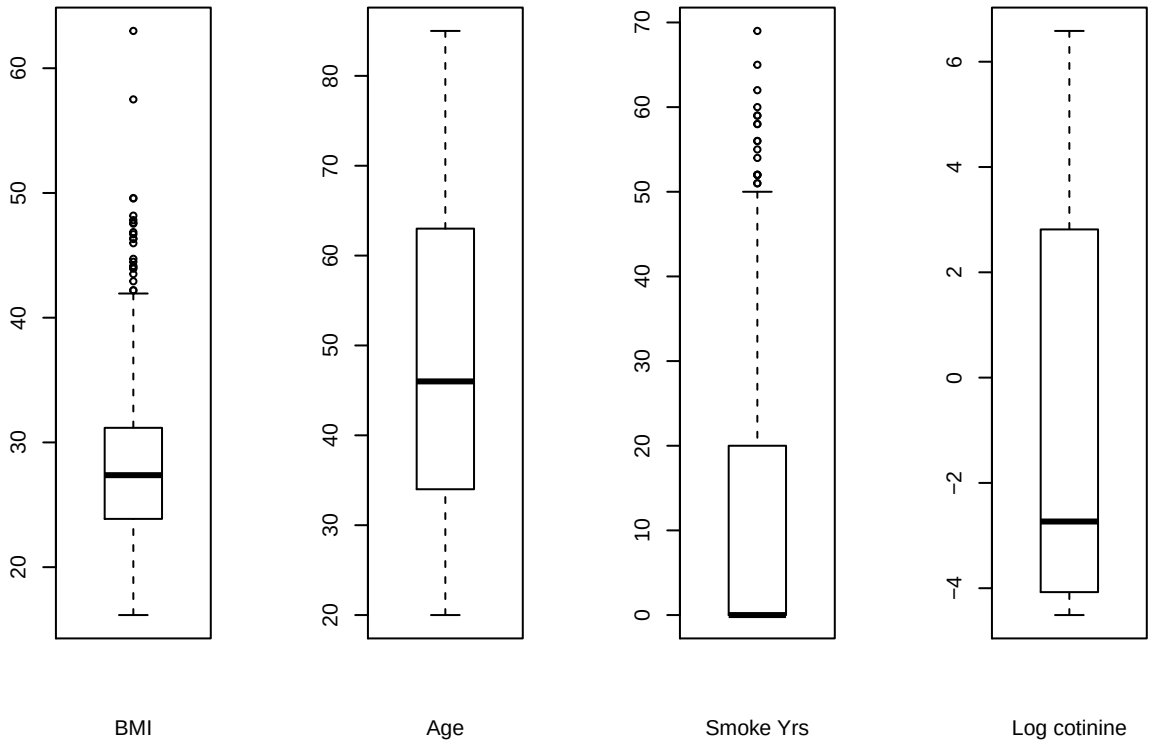
```
##  3rd Qu.:0.8000   3rd Qu.:31.17
##  Max.   :5.5000   Max.   :62.99
##      ageyrs         yrssmoke          smokenow     ln_lbxcot
##  Min.   :20.00   Min.   : 0.0   Non-Smoker:664   Min.   :-4.5099
##  1st Qu.:34.00   1st Qu.: 0.0   Smoker    :200   1st Qu.:-4.0745
##  Median :46.00   Median : 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
```

7.2 Boxplots




0
20000
40000
60000
80000
120000
PCB 7
0
50000
100000
150000
PCB 8
0
20000
40000
60000
80000
120000
PCB 9
0
50
100
150
PCB 10
0
200
400
600
800
PCB 11
0
10
20
30
40
Furan 1
0
5
10
15
20
25
30
Furan 2
0
10
20
30
Furan 3
0
50
100
150
200
Furan 4


Dioxin 1
Dioxin 2
Dioxin 3
WBC Cnt
lymph %
mono %
eosin %
baso %
neutro %



7.3 Outlier Entries

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

```
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) {
```

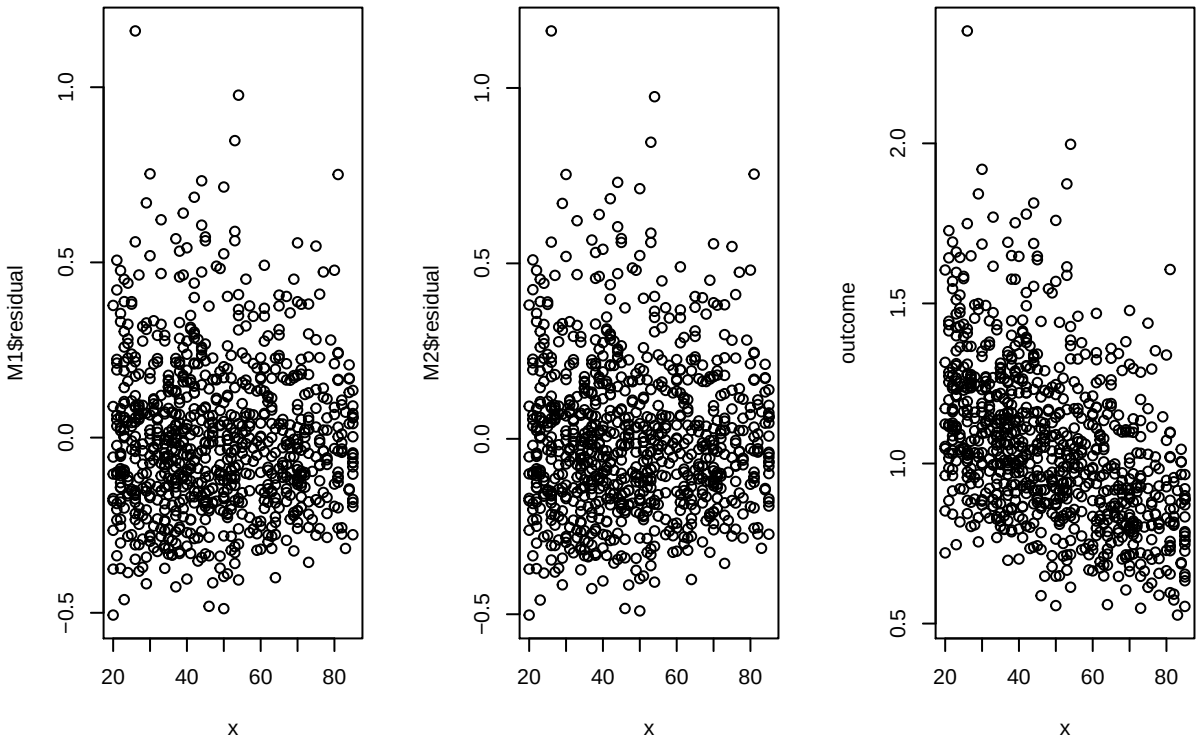
```
  max_WBC_idx[c-19] = which.max(pollutant_mat[, c])
}
max_WBC_idx
```

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

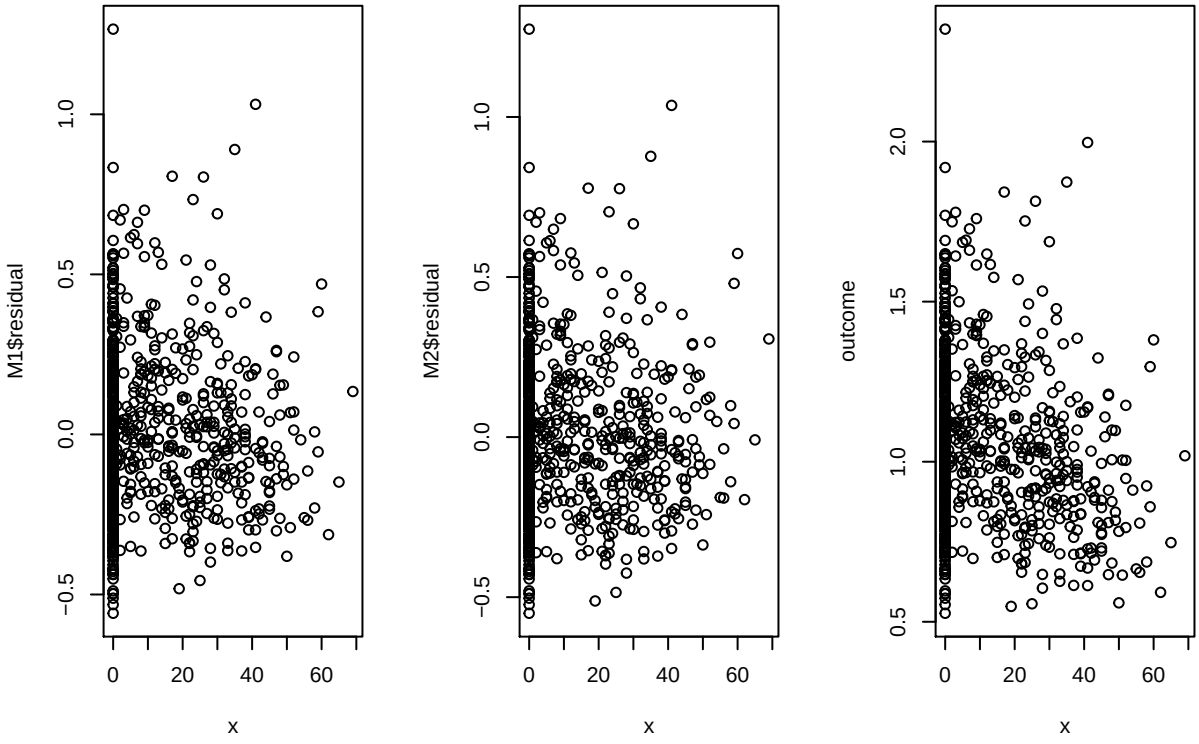
7.4 AvPlots

```
# 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)
}
```

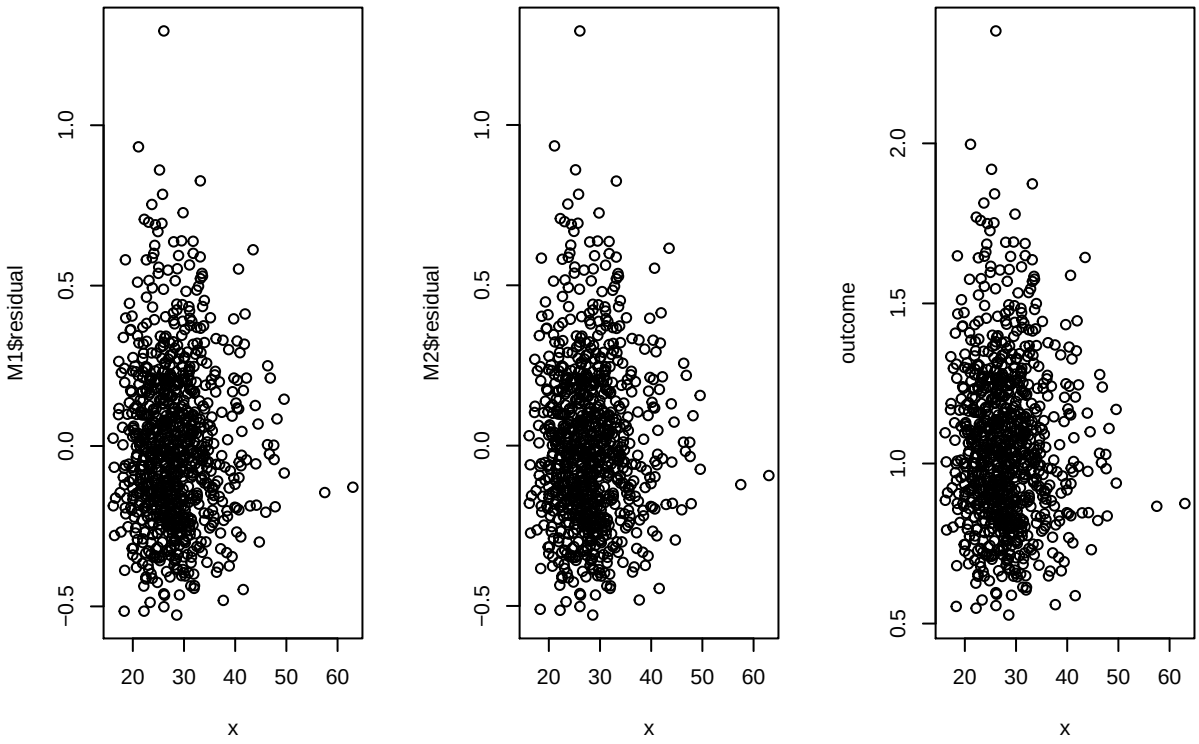
```
## [1] "residual for M1:  0.224172364185412"
## [1] "residual for M2:  0.22429269961392"
```



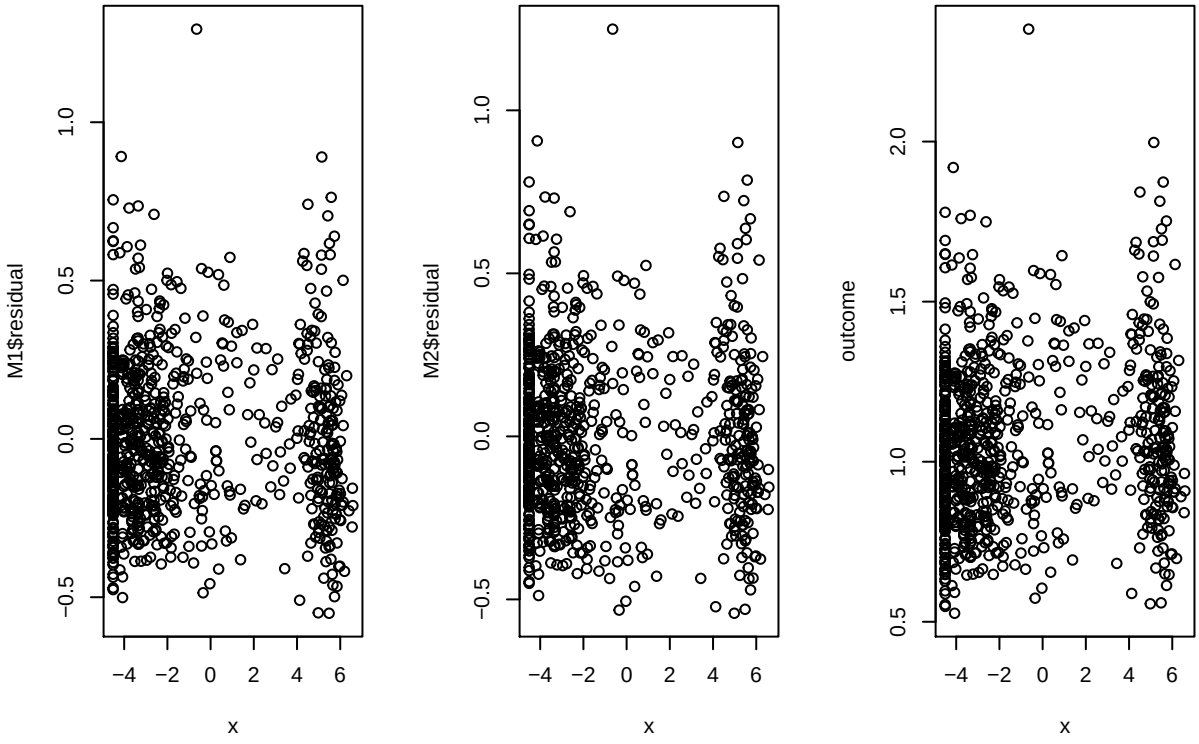

```
## [1] "residual for M1:  0.246320733146214"
## [1] "residual for M2:  0.245622720856213"
```



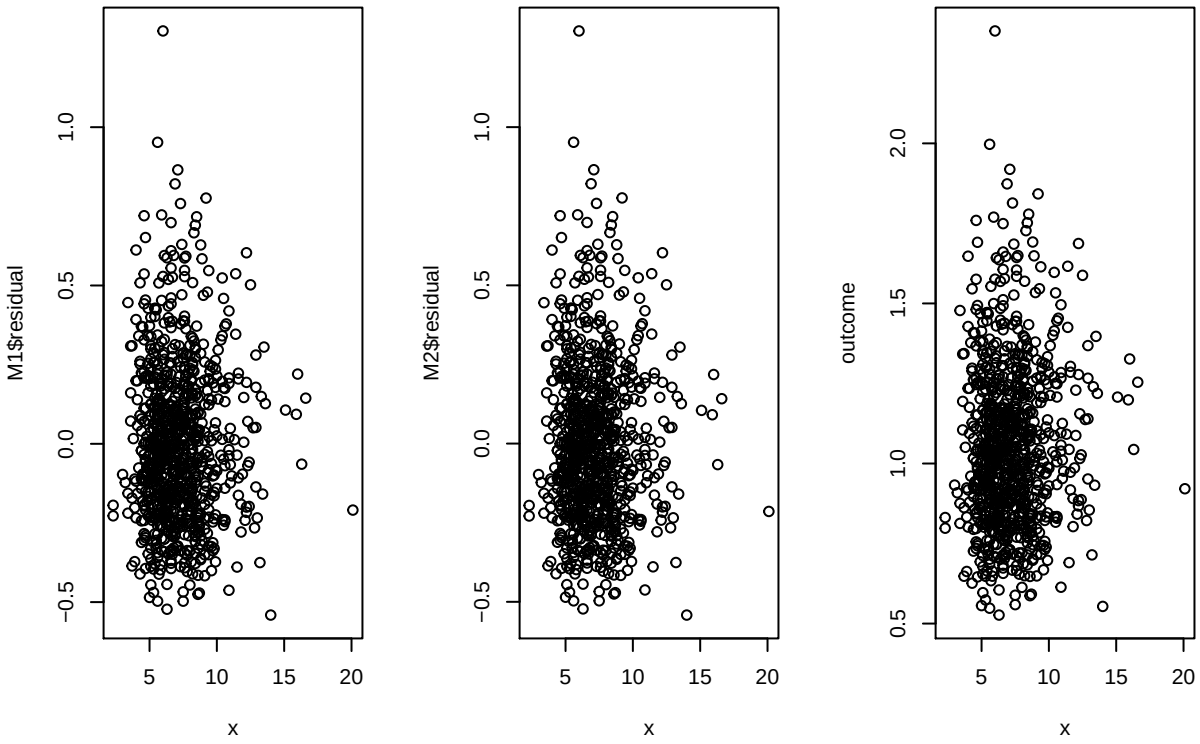

```
## [1] "residual for M1:  0.250228706427173"
## [1] "residual for M2:  0.25036248052387"
```



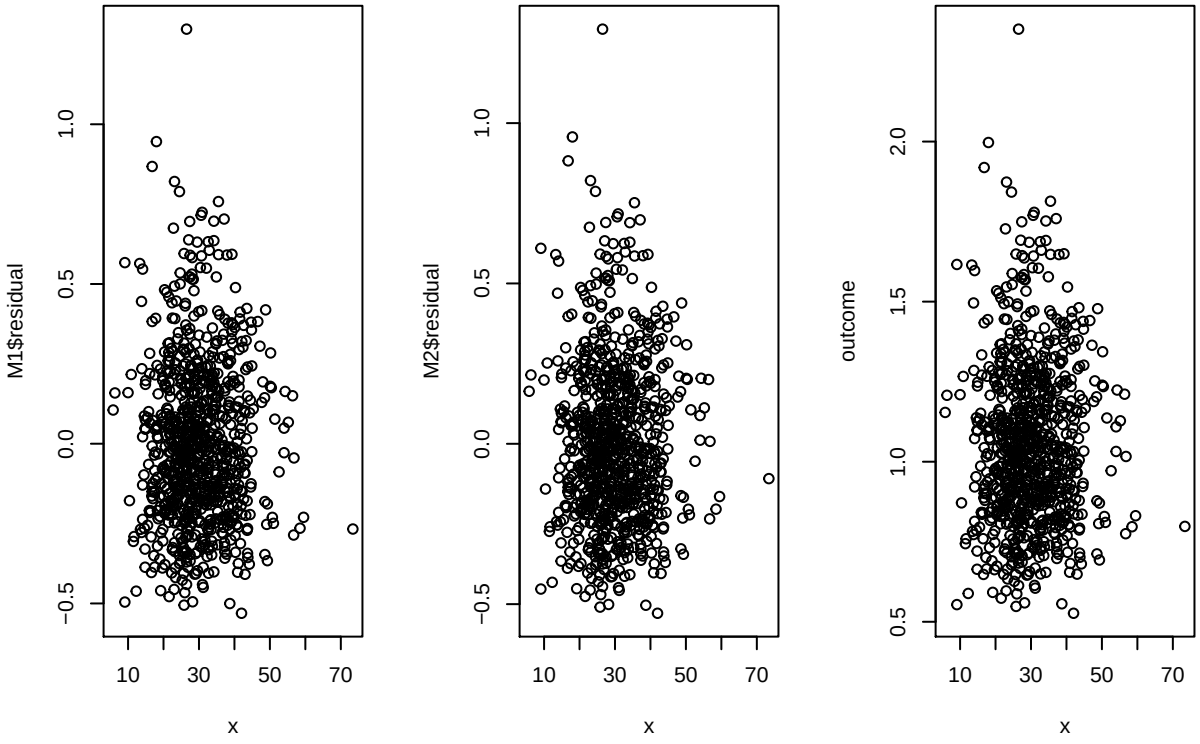

```
## [1] "residual for M1:  0.248212063673837"
## [1] "residual for M2:  0.24710732733351"
```



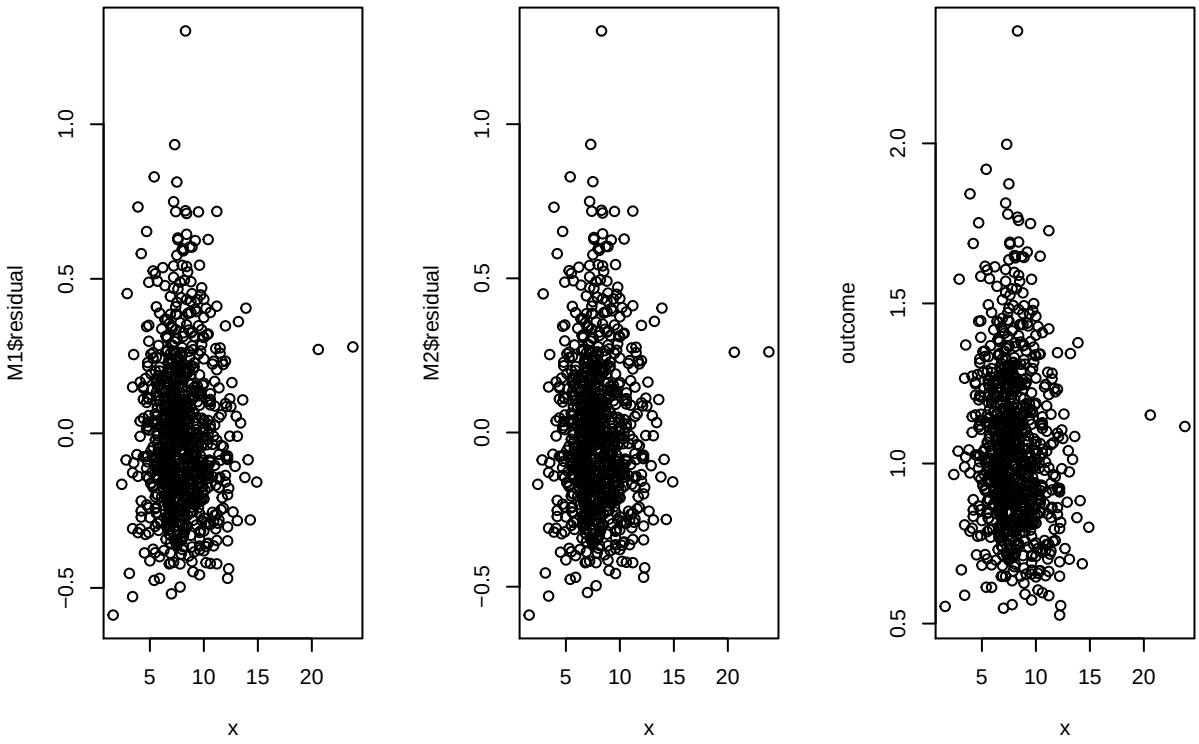

```
## [1] "residual for M1:  0.250065445847753"
## [1] "residual for M2:  0.250210403543218"
```



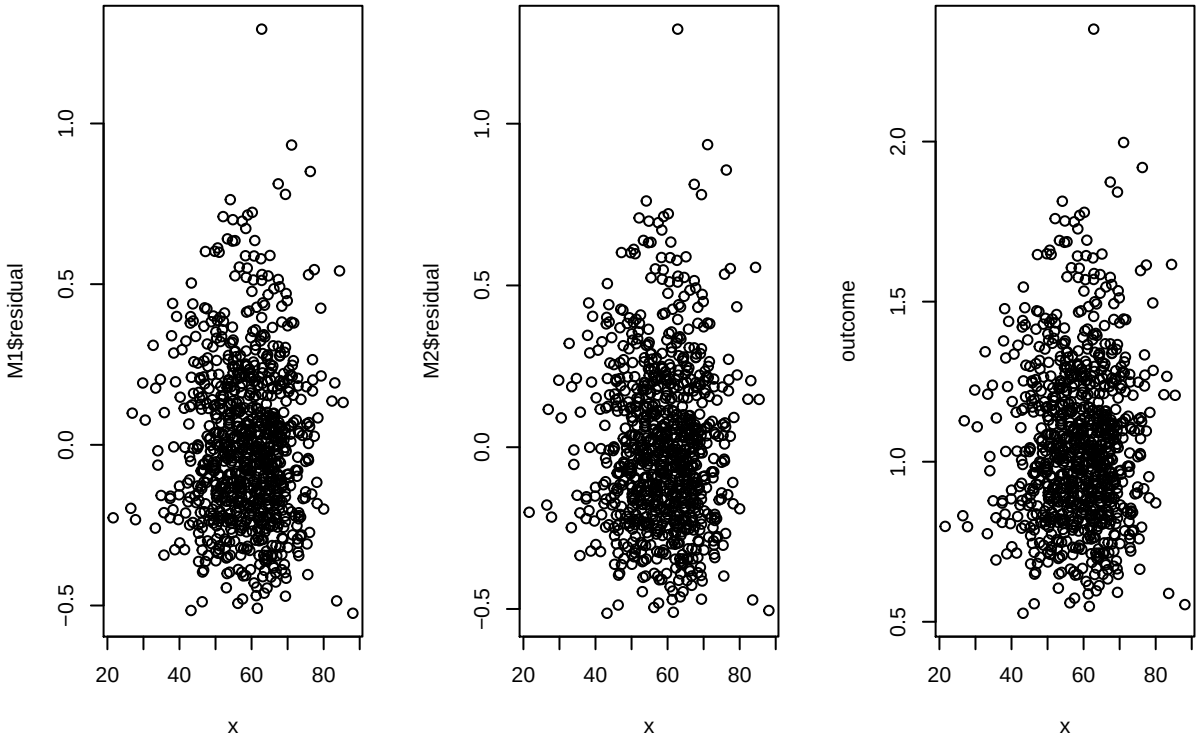

```
## [1] "residual for M1:  0.250373616826691"
## [1] "residual for M2:  0.250255208638358"
```



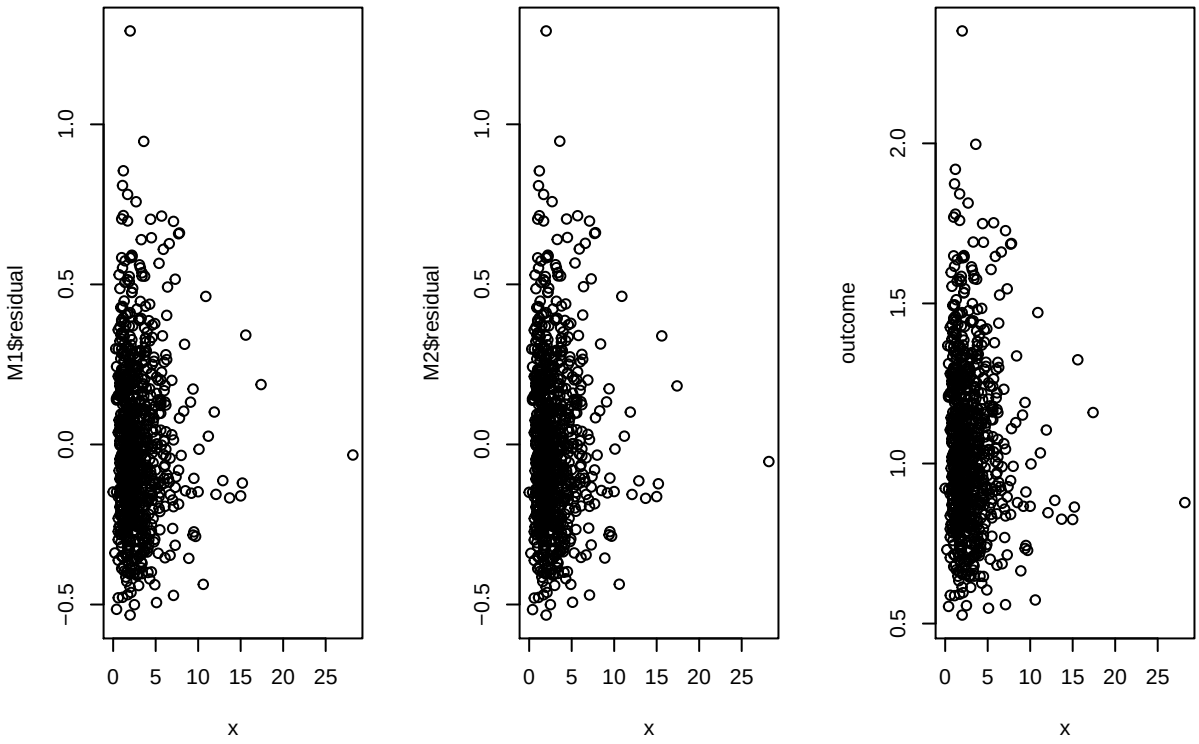

```
## [1] "residual for M1:  0.248704466454944"
## [1] "residual for M2:  0.248847192837983"
```



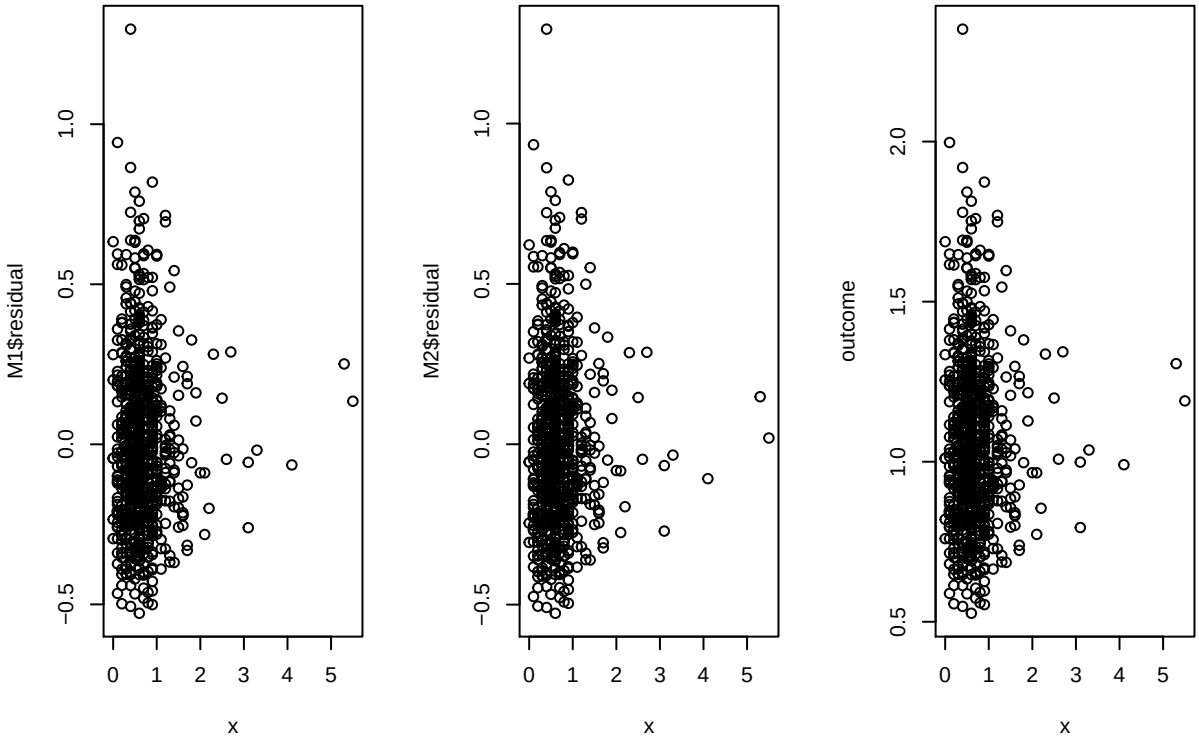

```
## [1] "residual for M1:  0.25026710930793"
## [1] "residual for M2:  0.250393729526099"
```




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




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

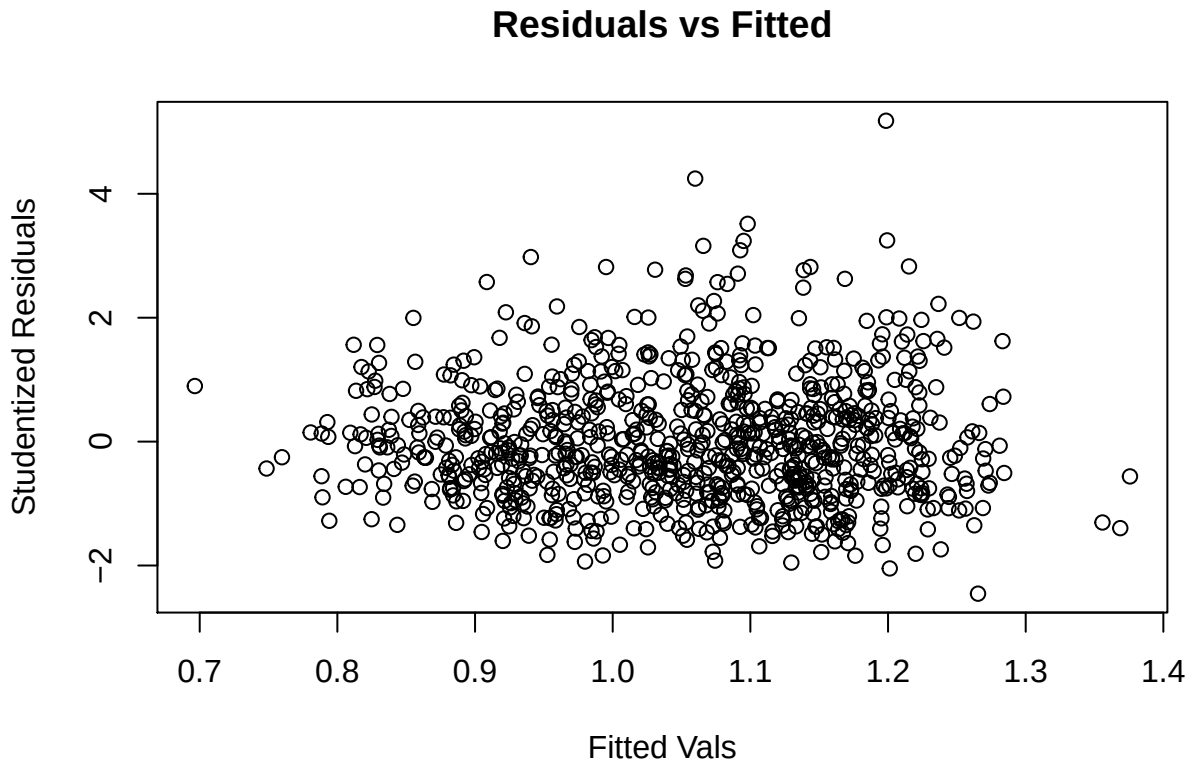



```
# # Judy's work Part 2
# # testing non-linearity in MLR
# library(car)
# M <- lm (length ~ ., data=pollutants)
```

```r
# avPlots(M, main="Added-Variable Plot")
```

7.5 Residuals vs Fitted plot

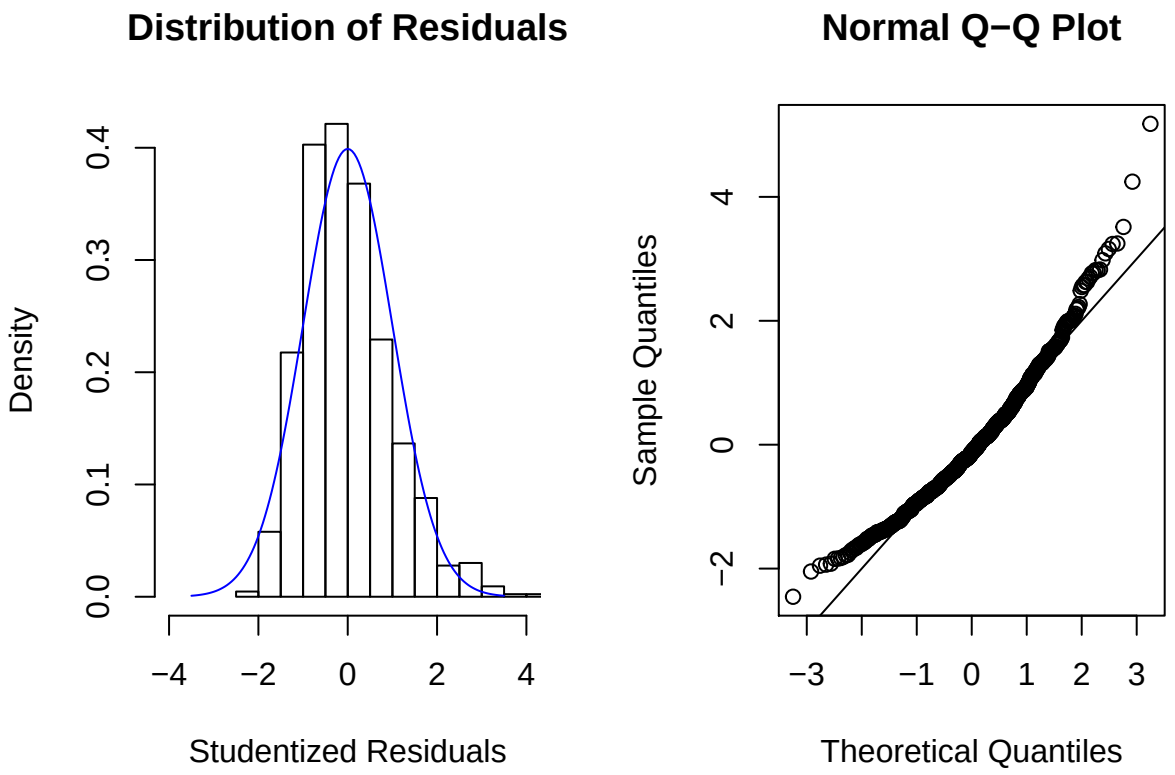
```r
# Heteroskedasticity
## fit model
Mh <- lm(length ~ . - smokenow - race_cat
         - edu_cat - male , data = pollutants)
## residuals
res1 <- resid(Mh) # raw residuals
stud1 <- res1/(sigma(Mh)*sqrt(1-hatvalues(Mh))) # studentized residuals

## plot of studentized residuals vs fitted values
plot(stud1~fitted(Mh),
     xlab="Fitted Vals",
     ylab="Studentized Residuals",
     main="Residuals vs Fitted")
```



7.6 Histograms and QQ plot

```r
par(mfrow = c(1, 2))
## plot distribution of studentized residuals
hist(stud1,breaks=12,
     probability=TRUE,xlim=c(-4,4),
     xlab="Studentized Residuals",
     main="Distribution of Residuals")
grid <- seq(-3.5,3.5,by=0.05)
lines(x=grid,y=dnorm(grid),col="blue") # add N(0,1) pdf
```

```
## qqplot of studentized residuals
qqnorm(stud1)
abline(0,1)
```



7.7 Model Summaries

comments by Estella: !need to revise, now we only have 3 models, see result above for more information

7.7.1 Models Selected with Interactions

```
# stepwiseB_Adjusted R2
# summary(cv_ridge_model1)
# summary(MBIC_Interaction)
```

7.7.2 Models after VIF Selection

```
# summary(MAIC_reduced)
# summary(MBIC_reduced)
```