Predictive Analysis of the Air Quality of Shunyi District in Beijing, China based on $PM_{2.5}$ Levels

Sean Torres and Anusia Edward

Shiley-Marcos School of Engineering, University of San Diego

Abstract

Air pollution in the form of fine particulate matter, $PM_{2.5}$, is contributing to respiratory related illnesses and deaths. The purpose of this study is to address this growing public health concern in the Shunyi District of Beijing, China by examining four predictive models. For this study, it was hypothesized that the top five predictors would be the actual pollutants found in the atmosphere: PM_{10}, SO_2, NO_2, CO, and O_3. To carry out this hypothesis the following models were examined: ordinary least square regression (OLS) model, partial least squares (PLS) model, Random Forest model, and Elastic Net model. Subsequently it was hypothesized that the best model will be the Elastic Net model with an R^2 value of 0.8. The data was preprocessed to correct for multicollinearity, near zero variance predictors, skewness, and outliers. The Elastic Net model was found to be the best model with the top five predictors as PM_{10}, CO, NO_2, SO_2, and WSPM.

Keywords: air pollution, $PM_{2.5}$, OLS model, PLS model, Random Forest model, and Elastic Net model

Table of Contents

Introduction **4**

Background and Practical Implications 4

Purpose, Objectives, Hypothesis, and Justification of Present Study 5

Methods **6**

Data Collection and Sample Characteristics 6

Exploratory Data Analysis 7

Data Wrangling, Pre-Processing, and Data Splitting 10

Modeling **11**

Model Building Strategies 11

Model Performance Evaluation Metrics and Hyperparameter Tuning 12

Results **12**

Model Performance and Final Model Selection 12

Discussion **13**

References **15**

Appendix A **18**

Appendix B **23**

Introduction

Background and Practical Implications

Air pollution accumulates through a combination of both natural and man-made sources. Some natural sources that contribute to air pollution include gasses emitted from agricultural practices, smoke from wildfires, or ash from volcanic eruptions. Man-made sources that contribute to air pollution include pollutants emitted from vehicles, coal-fueled power plants, fumes from chemical production, and gas to heat homes (Manisalidis et al., 2020). Air pollution is composed of a collection of harmful substances that can cause chronic health issues as well as premature mortality. The specific harmful substances that make up air pollution include noxious gasses, ground-level ozone, sulfur oxides, volatile organic compounds, different carbon forms, and fine particulate matter. Fine particulate matter, $PM_{2.5}$, is considered the most harmful pollutant to one's health as it is closely associated with causing premature death (Waidyatillake et al., 2021). $PM_{2.5}$ is found in dust, dirt, smoke, and liquid droplets. A large portion of particulate matter is emitted by diesel vehicles and coal-fired power plants. $PM_{2.5}$ is particularly hazardous as the particles are so small that they lodge deep into the lungs causing the lungs to become aggravated and inflamed leading to significant health risks (Xing et al., 2016).

In the past two decades, there has been a rapid increase in the number of deaths around the world as a result of air pollution, especially in China (Zheng et al., 2021). China’s air quality is ranked as one of the lowest among countries worldwide based on its Environmental Sustainability Index value. One of the most affected locations in China is the Shunyi District of Beijing (Wang et al., 2018). This particular district in China is known for having a major international airport, two metro lines, as well as issues with unregulated factories, which heavily contribute to the air pollution seen in Shunyi (Economy, 2014). Short term, as well as long term,

exposure to high levels of air pollution have been known to increase the risk of respiratory infections, heart disease, and lung cancer (Kloog et al., 2013). The rising deaths in Shunyi due to air pollution highlight the importance in further studying this public health issue (Wang et al., 2018).

Purpose, Objectives, Hypothesis, and Justification of Present Study

The purpose of this study is to address the growing issue of poor air quality and its negative effects on health outcomes by analyzing the Shunyi District of Beijing, China. This study's objective, therefore, is to use secondary data to develop a predictive model aimed at predicting air quality in terms of $PM_{2.5}$ levels based on units of time (month, year, hour), corresponding pollutants such as PM_{10}, sulfur dioxide (SO_2), nitrogen dioxide (NO_2), carbon monoxide (CO), Ozone (O_3), temperature (TEMP), pressure (PRES), dew point temperature (DEWP), and wind speed (WSPM). In regards to time, month and year can be helpful for predicting $PM_{2.5}$ levels as the use of residential fireplaces and woodstoves during the colder seasons can contribute to an increase in $PM_{2.5}$ levels compared to warmer months during the year. Additionally, past studies have found that $PM_{2.5}$ levels are higher from 6 a.m. to 10 p.m. as those are peak hours that individuals are moving around in modes of transportation that involve the emission of particles that contribute to the presence of $PM_{2.5}$ in the atmosphere (Srimuruganandam et al., 2010). This observation provides insight as to why time was selected as a potential predictor for $PM_{2.5}$. Past research has found that the presence of PM_{10} can also help with predicting levels of $PM_{2.5}$ (Biancofiore et al., 2017). Furthermore, the concentrations of the following are present within the atmosphere due to the emissions of fossil plants and fuel: SO_2, NO_2, CO, and O_3 (Han et al., 2021). The presence of these compounds within the atmosphere may be indicative of the presence of $PM_{2.5}$, which is also caused by emissions of fossil plants and

fuel. Temperature has been found to affect $PM_{2.5}$ in relation to cardiovascular mortality (Li et al., 2015). Wind speed and pressure share a chemical relationship with temperature. Thus, these three characteristics were taken into account as potential predictors for $PM_{2.5}$. Therefore the following variables can help with predicting levels of $PM_{2.5}$: time (month, year, hour), corresponding pollutants such as PM_{10}, SO_2 (ug/m^3), NO_2 (ug/m^3), CO (ug/m^3), O_3 (ug/m^3), temperature (℃), pressure (hPa), dew point temperature (℃), and wind speed (m/s). For this study, it was hypothesized that the top five predictors would be the actual pollutants found in the atmosphere: PM_{10}, SO_2, NO_2, CO, and O_3. To carry out this hypothesis the following models were examined: ordinary least square regression (OLS) model, partial least squares (PLS) model, Random Forest model, and Elastic Net model. Subsequently it was hypothesized that the best model will be the Elastic Net model with an R^2 value of 0.8.

Methods

Data Collection and Sample Characteristics

This study is a retrospective data analysis based on the raw data which was sourced from the UCI Machine Learning Repository. The raw data was initially obtained as an unzipped CSV file that was adjusted in R-studio for further use and exploration. R-studio was used as the primary system of analysis for this study. It should be noted that for the purpose of this study the sample population being observed is Shunyi District of Beijing, China. The initial dataset consisted of 35,064 records and sixteen attributes which include the following: year, month, day, hour, $PM_{2.5}$, PM_{10}, SO_2, NO_2, CO, O_3, TEMP, PRES, DEWP, RAIN, WSPM, and station. All data pertaining to this dataset was collected from the Beijing Municipal Environmental Monitoring Center from March 1, 2013 to February 28, 2017. The year, month, and day variables are all numeric variables indicating the date on which the data was collected. The hour attribute is a

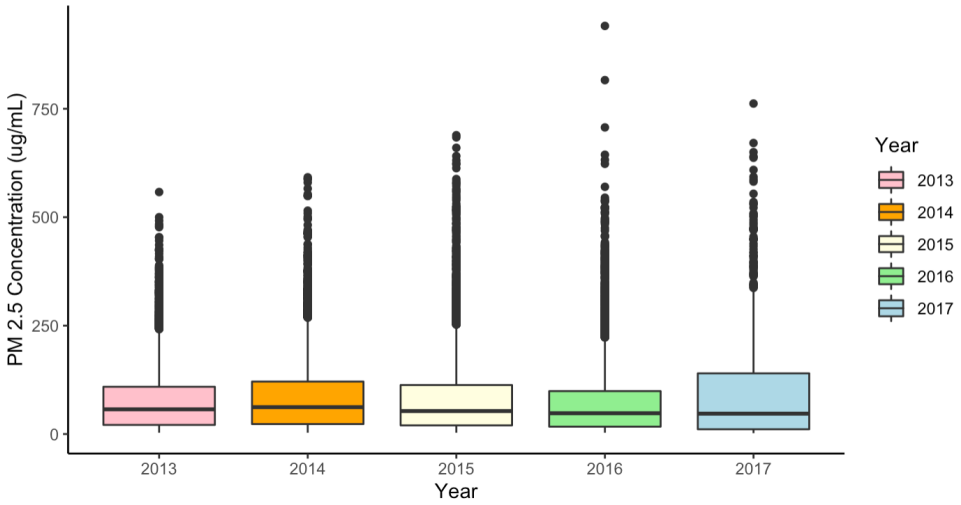
numeric variable and refers to the hour at which the data was collected on a specified day. The hour variable had a range of 0 indicating 12:00 a.m. to 23 indicating 11:00 p.m. PM_{10} is a numeric variable that refers to the concentration of large particulate matter in the air and is recorded in ug/m^3. The range for this variable is 2 to 999 ug/m^3, where values over 430 indicate hazardous air quality. SO_2, NO_2, CO, and O_3 are continuous variables that indicate the concentrations of each of these compounds in the air. SO_2 has a range of 0.286-239 ug/m^3, NO_2 has a range of 2-258 ug/m^3, CO has a range of 0-10000 ug/m^3, and O_3 has a range of 0.2142-351.7 ug/m^3. The TEMP variable is a continuous variable with a range of -16.8℃ to 40.6℃. PRES is a continuous variable with a range of 988-1043 hPa. DEWP is a continuous variable with a range of -36℃ to 27.5℃. RAIN is a continuous variable with a range of 0 to 37.3 mm. WSPM is a continuous variable with a range of 0 to 12.8 m/s. "Station" is a categorical variable indicating the name of the air-quality monitoring site, which is Shunyi. The target variable, $PM_{2.5}$ (ug/m^3), is a numeric variable of the concentration of fine particulate matter.

Exploratory Data Analysis

Prior to performing data cleaning, relationships between variables were examined through the use of box plots. The first box plot observed the relationship of $PM_{2.5}$ concentrations across the years 2013-2017 (Figure 1). The median values for each of the years (2013-2017) were found to be: 58, 62, 54, 49, and 47.5 ug/m^3. The year with the highest median $PM_{2.5}$ concentration was 2014 (62 ug/m^3), and the year with the lowest median $PM_{2.5}$ concentration was 2017 (47.5 ug/m^3). This may be the case as the data regarding 2017 was only from the first two months of the entire year, whereas the previous years had data covering all twelve months.

Figure 1.

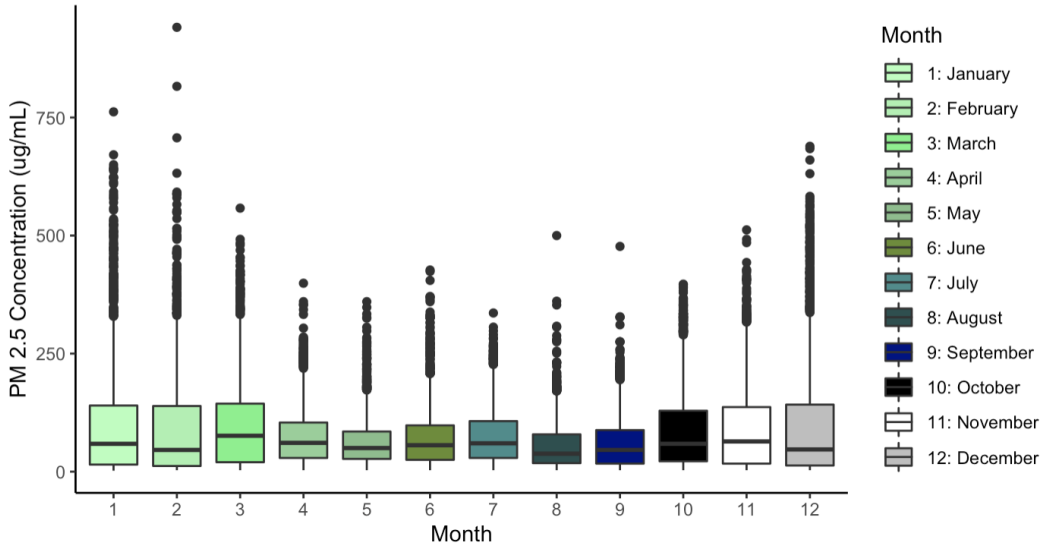
PM2.5 levels for 2013-2017 in the Shunyi District



Additionally, $PM_{2.5}$ concentrations were also observed across the twelve months from 2013-2017 (Figure 2). Median $PM_{2.5}$ concentrations across the months were recorded as follows: 60, 48, 76, 61, 50, 56, 59, 39, 46.5, 59, 64, and 46 ug/m^3. March had the highest median concentration (76 ug/m^3), while August had the lowest median concentration (39 ug/m^3). Late November through early March marks the winter season in China, with the maximum average daily temperature typically below 7°C (Monteiro et al., 2013). The majority of China's central heating system, used during the winter months, is generally a coal-based operation (Xiao et al., 2015). Therefore, the release of $PM_{2.5}$ would be expected to be higher during these months, as coal is a high contributor to $PM_{2.5}$ levels (Manisalidis et al., 2020). Additionally, the rainy season in China is generally from May to September, and rain reduces the amount of air pollution present within the air as it causes air pollutants to be washed away (Zhang et al., 2021). Based on this phenomenon, it is plausible that August had the lowest $PM_{2.5}$ levels.

Figure 2.

PM2.5 Levels observed in the Shunyi District by Month from 2013 to 2017



Correspondingly, descriptive statistics for $PM_{2.5}$, PM_{10}, SO_2, NO_2, CO_2, O_3, TEMP, PRES, DEWP, RAIN, and WSPM were analyzed (Table 1). The average $PM_{2.5}$ value for Shunyi District from 2013 to 2017 was found to be 79.49 ug/m^3, indicating that the air quality level for Shunyi is unhealthy for at-risk individuals based on the air quality index (Xing et al., 2016). Additionally, it should be noted that the PM_{10} average value was found to be 98.72 ug/m^3, which further confirms that on average, citizens of Shunyi were faced with unhealthy air quality conditions from 2013 to 2017. It is also important to note the maximum values for each of the variables, as these values provide insight on the extent to which individuals of Shunyi have been exposed to dangerous air pollutants. Most notably, $PM_{2.5}$ and PM_{10} have a maximum value of 941 and 999 ug/m^3, which indicate extremely unhealthy air quality for all exposed individuals.

Table 1

Descriptive Statistics for Quantitative Variables

	Mean	Median	Minimum	Maximum	IQR	25th Percentile	75th Percentile
$PM_{2.5}$ (ug/m^3)	79.49	55	2	941	93	19	112
PM_{10} (ug/m^3)	98.74	77	2	999	107	31	138
SO_2 (ug/m^3)	13.572	5	0.2856	239	15	2	17
NO_2 (ug/m^3)	43.91	37	2	258	43	19	62
CO (ug/m^3)	1187	800	100	10000	1100	400	1500
O_3 (ug/m^3)	55.2013	43	0.2142	351.7164	67	10	77
TEMP (°C)	13.39	14.4	-16.8	40.6	20.2	3	23.2
PRES (hPa)	1013	1013	988	1043	16	1005	1021
DEWP (°C)	2.465	3.1	-36	27.5	23.9	-8.8	15.1
RAIN (mm)	0.06109	0	0	37.3	0	0	0
WSPM (m/s)	1.808	1.5	0	12.8	1.3	1	2.3

Abbreviations: PM, Particulate matter; SO2, sulfur dioxide; NO2, Nitrogen dioxide; CO, carbon monoxide; O3, ozone; TEMP, temperature; PRES, pressure; DEWP, Dewpoint temperature; WSPM, windspeed

Data Wrangling, Pre-Processing, and Data Splitting

In order to ensure that there was no data leakage, the first step attempted was to split the dataset using the *createDataPartition* of the "caret" package; however, the function would not permit splitting the data when the outcome variable had missing values. Therefore, the missing values for only the outcome variable were imputed using K-nearest neighbors (KNN). KNN was used to impute these values because it utilizes similar records in the dataset to calculate appropriate values for missing data (Acuna & Rodriguez, 2004). Next, the data was split using a stratified random sampling approach in order to ensure that the proportions of the target variable, $PM_{2.5}$, was proportional within the sets, and to ensure that there was no further data leakage. This was carried out using the *createDataPartition* function in the "caret" package. The outcome and predictor variables for the train and test sets were then split, and the missing values for the predictors were imputed using KNN. Near zero variance variables were removed from the predictor variables as they, by definition, have low predictive ability (Kumar & RamaSree, 2015). Then, boxplots were produced to check for outliers (Figure A1). PM_{10}, SO_2, NO_2, CO, O_3, and WSPM had multiple outliers. Histograms were also produced to visualize the distributions of each of the predictors (Figure A2). PM_{10}, SO_2, NO_2, CO, O_3, and WSPM had right skewed

distributions, while DEWP had a left skewed distribution. In order to handle the outliers as well as skewness, the data were preprocessed using the following transformations: box-cox, centering, and scaling. All three transformations were carried out using the *preProcess* function of the "caret" package. After these transformations, the concerns of skewness and outliers were eliminated as seen in Figures A3 and A4. Finally, multicollinearity was adjusted for using the *findCorrelation* function of the "caret package" (Appendix B). Pearson's correlation coefficient was used to determine this, since the data were continuous and normal. The cutoff threshold for correlation was set to 0.75. The *createFolds* function was used to split the data into groups for use of resampling at each of the iterations. The resample method was determined to be cross-validation as seen under the "ctrl" variable (Appendix B).

Modeling

Model Building Strategies

The base model used to test the data was OLS. OLS is easy to interpret, and it is appropriate to utilize when working with quantitative independent and dependent variables. Aided by PCA, the OLS model can be a good candidate to further understand relationships within the dataset. However, the OLS model struggles with multicollinearity in which case, the PLS regression may be more favorable. Although multicollinearity was previously adjusted for, complete correlation between other predictors can not be ensured; therefore, PLS was chosen to accommodate for the presence of correlated predictors. The next model selected was the Random Forest model. Random Forest was considered as it takes into account high-dimensionality. The last model chosen was the Elastic Net model. The Elastic Net model is a model that addresses issues with correlated predictors. Furthermore, it also takes into account concerns of overfitting, underfitting, and variance, which helps reduce the overall prediction error of the model.

Model Performance Evaluation Metrics and Hyperparameter Tuning

The metrics primarily used to determine the success of the models were R^2 and root mean square error (RMSE). R^2 indicates the variation in the output variable as a result of the predictor variables. A high R^2 value demonstrates that the independent variables are effectively predicting the outcome variable. RMSE analyzes the standard deviation of residuals. A low value for RMSE indicates that the model is a good fit for the data. ROC and AUC were not used, as these measures are not appropriate for regression models and are better suited for classification models such as logistic regression. The top five predictor variables for each of the models were determined using the *varImp* factor, which predicts the variable importance depending on the specific type of model used.

For the Random Forest and Elastic Net models, hyperparameter tuning was implemented to maximize the model's performance. The key hyperparameters that can be tuned for the Elastic Net model include alpha and lambda. Alpha can be tuned on a range of 0 to 1, while lambda can be tuned from -10 to 10. The Elastic Net alpha was set to one, while lambda was set to zero. The hyperparameters that can be tuned for the Random Forest model include the following: *num.trees, mtry,* and *min.node.size* under the "TuneRanger" package. The range for *num.trees* is 1 to 2000, while the range for *mtry* is 0 to 1, and the range for *min.node.size* is 0 to 1. The Random Forest was tuned with a value of 500 trees. Alterations in these hyperparameters can help increase overall model performance (Probst et al., 2017). OLS and PLS did not have any tuning parameters to increase performance any further.

Results

Model Performance and Final Model Selection

The OLS model had an RMSE of 45.3 and an R^2 value of 0.6951. The top five predictors for the OLS model, based on the scaled *varImp* values are: PM_{10} (100), CO (41.41), SO_2 (8.253), PRES (7.519), and O_3 (4.986). The PLS model had an RMSE value of 43.25 and an R^2 value of 0.6993. The top five predictors for the PLS model, based on the scaled *varImp* values are: PM_{10} (100), CO (81.67), NO_2 (68.05), SO_2 (50.02), and WSPM (30.03). The Random Forest model had an RMSE value of 91.22 and an R^2 value of 6.88e-05. The top five predictors for the Random Forest model, based on the scaled *varImp* values are: PM_{10} (100), CO (61.72), TEMP (41.97), NO_2 (34.03), and month (26.89). The Elastic Net model had an RMSE value of 42.92 and an R^2 value of 0.7039. The Elastic Net model was identified as the final model as it had the highest R^2 value (0.7039) and the lowest RMSE value (43.25). The top five predictors for the Elastic Net model, based on the scaled *varImp* values are: PM_{10} (100), CO (76.61), NO_2 (51.91), SO_2 (25.66), and WSPM (10.86).

Table 2.

Evaluation of OLS, PLS, Random Forest and Elastic Net models.

Model	RMSE	R^2	MAE
OLS	45.3	0.6951	31.54
PLS	43.25	0.6993	30.86
Random Forest	91.22	6.88E-05	58.97
Elastic Net	42.92	0.7039	30.64

Abbreviations: OLS, Ordinary Least Squares Regression; PLS, partial least squares regression; RMSE, root means square error; MAE, mean absolute error

Discussion

The results obtained for the final model selection indicate that the Elastic Net model is the best model to predict $PM_{2.5}$ levels in the Shunyi District of Beijing, China. Although the PLS model came as a close second, it makes sense that the Elastic Net model was able to outperform it, as the Elastic Net model has been found to have more overall stability and dependability with

its regression coefficients (Liu and Li, 2017). The OLS model came in third as overall, OLS does not consider multicollinearity, which may have been present among the pollutant predictors: PM_{10}, SO_2, NO_2, CO, and O_3. Lastly, the Random Forest model performed the worst. This makes sense, as Random Forest is usually preferred for classification models.

When observing which predictors best predict $PM_{2.5}$ levels, the top five predictors for the Elastic Net model were observed to be: PM_{10}, CO, NO_2, SO_2, and WSPM. Past studies have found that CO, NO_2, and SO_2 are pollutants in the air as a byproduct of fossil fuels (Han et al., 2021). In addition, since $PM_{2.5}$ is also a byproduct of fossil fuels, the presence of the following compounds CO, NO_2, SO_2 can indicate the potential presence of $PM_{2.5}$ in the atmosphere (Han et al., 2021). WSPM was not hypothesized to be a top predictor for $PM_{2.5}$. However, WSPM has the potential to affect air pollutant concentration via diffusion of air pollutants, which is why WSPM may have been observed as one of the top predictors for $PM_{2.5}$ (Perez et al., 2021).

Future studies can look to further examine the relationship of WSPM as an air pollutant predictor. In addition, other districts in China can be cross examined to see if the predictors for $PM_{2.5}$ change by location. Furthermore, another recommended model to explore would be the Autoregressive integrated moving average (ARIMA) model. The ARIMA model is a statistical model that can be used to evaluate time series data (Li et al., 2021). This may be beneficial, as the data observed in this study looks at records over the course of a four year span.

References

Acuna, E., & Rodriguez, C. (2004). *The treatment of missing values and its effect on classifier accuracy. In Classification, clustering, and data mining applications* (pp. 639-647). Springer, Berlin, Heidelberg.

Biancofiore, F., Busilacchio, M., Verdecchia, M., Tomassetti, B., Aruffo, E., Bianco, S., & Di Carlo, P. (2017). *Recursive neural network model for analysis and forecast of PM10 and PM2. 5.* Atmospheric Pollution Research, 8(4), 652-659.

Economy, E. (2014). *Environmental governance in China: State control to crisis management.* Daedalus, 143(2), 184-197.

Han, P., Mei, H., Liu, D., Zeng, N., Tang, X., Wang, Y., & Pan, Y. (2021). *Calibrations of low-cost air pollution monitoring sensors for CO, NO2, O3, and SO2*. Sensors, 21(1), 256.

Kloog, I., Ridgway, B., Koutrakis, P., Coull, B. A., & Schwartz, J. D. (2013). *Long- and short-term exposure to PM2.5 and mortality: using novel exposure models.* Epidemiology (Cambridge, Mass.), 24(4), 555–561. https://doi.org/10.1097/EDE.0b013e318294beaa

Kumar, S., & RamaSree, R. J. (2015, January). Dimensionality reduction in automated evaluation of descriptive answers through zero variance, near zero variance and non frequent words techniques-a comparison. *In 2015 IEEE 9th International Conference on Intelligent Systems and Control* (ISCO) (pp. 1-6). *IEEE*.

Li, Y., Ma, Z., Zheng, C., & Shang, Y. (2015). *Ambient temperature enhanced acute cardiovascular-respiratory mortality effects of PM2. 5 in Beijing, China.* International journal of biometeorology, 59(12), 1761-1770.

Li YR, Han TT, Wang JX, Quan WJ, He D, Jiao RG, Wu J, Guo H, Ma ZQ. (2021) *Application*

of ARIMA Model for Mid- and Long-term Forecasting of Ozone Concentration. 42(7):3118-3126. doi: 10.13227/j.hjkx.202011237. PMID: 34212637.

Liu, W., & Li, Q. (2017). An Efficient Elastic Net with Regression Coefficients Method for Variable Selection of Spectrum Data. PloS one, 12(2), e0171122. https://doi.org/10.1371/journal.pone.0171122

Manisalidis, I., Stavropoulou, E., Stavropoulos, A., & Bezirtzoglou, E. (2020). *Environmental and Health Impacts of Air Pollution: A Review.* Frontiers in public health, 8, 14. https://doi.org/10.3389/fpubh.2020.00014

Monteiro, A., Carvalho, V., Góis, J. Sousa, C. (2013). *Use of "Cold Spell" indices to quantify excess chronic obstructive pulmonary disease (COPD) morbidity during winter (November to March 2000–2007): case study in Porto*. International Journal Biometeorology, 57, 857–870. https://doi.org/10.1007/s00484-012-0613-z

Pérez, I. A., García, M., Sánchez, M. L., & Pardo, N. (2021). Influence of Wind Speed on CO2 and CH4 Concentrations at a Rural Site. International journal of environmental research and public health, 18(16), 8397. https://doi.org/10.3390/ijerph18168397

Probst, P., Boulesteix, A. L., & Bischl, B. (2019). Tunability: importance of hyperparameters of machine learning algorithms. The Journal of Machine Learning Research, 20(1), 1934-1965.

Srimuruganandam, B., & Shiva Nagendra, S. (2010). *Analysis and interpretation of particulate matter – PM10, PM2.5 and PM1 emissions from the heterogeneous traffic near an urban roadway.* Atmospheric Pollution Research, 1(3), 184-194. https://doi.org/10.5094/apr.2010.024

Waidyatillake, N. T., Campbell, P. T., Vicendese, D., Dharmage, S. C., Curto, A., & Stevenson,

M. (2021). *Particulate Matter and Premature Mortality: A Bayesian Meta-Analysis.* International journal of environmental research and public health, 18(14), 7655. https://doi.org/10.3390/ijerph18147655

Wang, L., Zhang, F., Pilot, E., Yu, J., Nie, C., Holdaway, J., Yang, L., Li, Y., Wang, W., Vardoulakis, S., & Krafft, T. (2018). *Taking Action on Air Pollution Control in the Beijing-Tianjin-Hebei (BTH) Region: Progress, Challenges and Opportunities.* International journal of environmental research and public health, 15(2), 306. https://doi.org/10.3390/ijerph15020306

Xiao, Q., Ma, Z., Li, S., & Liu, Y. (2015). The impact of winter heating on air pollution in China. PloS one, 10(1), e0117311. https://doi.org/10.1371/journal.pone.0117311

Xing, Y. F., Xu, Y. H., Shi, M. H., & Lian, Y. X. (2016). *The impact of PM2.5 on the human respiratory system*. Journal of thoracic disease, 8(1), E69–E74. https://doi.org/10.3978/j.issn.2072-1439.2016.01.19

Zhang, Y., Zang, S., Shen, X., & Fan, G. (2021). Observed Changes of Rain-Season Precipitation in China from 1960 to 2018. International journal of environmental research and public health, 18(19), 10031. https://doi.org/10.3390/ijerph181910031

Zheng, S., Schlink, U., Ho, K. F., Singh, R. P., & Pozzer, A. (2021). *Spatial Distribution of PM2.5-Related Premature Mortality in China*. GeoHealth, 5(12), e2021GH000532. https://doi.org/10.1029/2021GH000532

Appendix A

Figure A1.

Boxplots of each of the predictor variables before transformations.

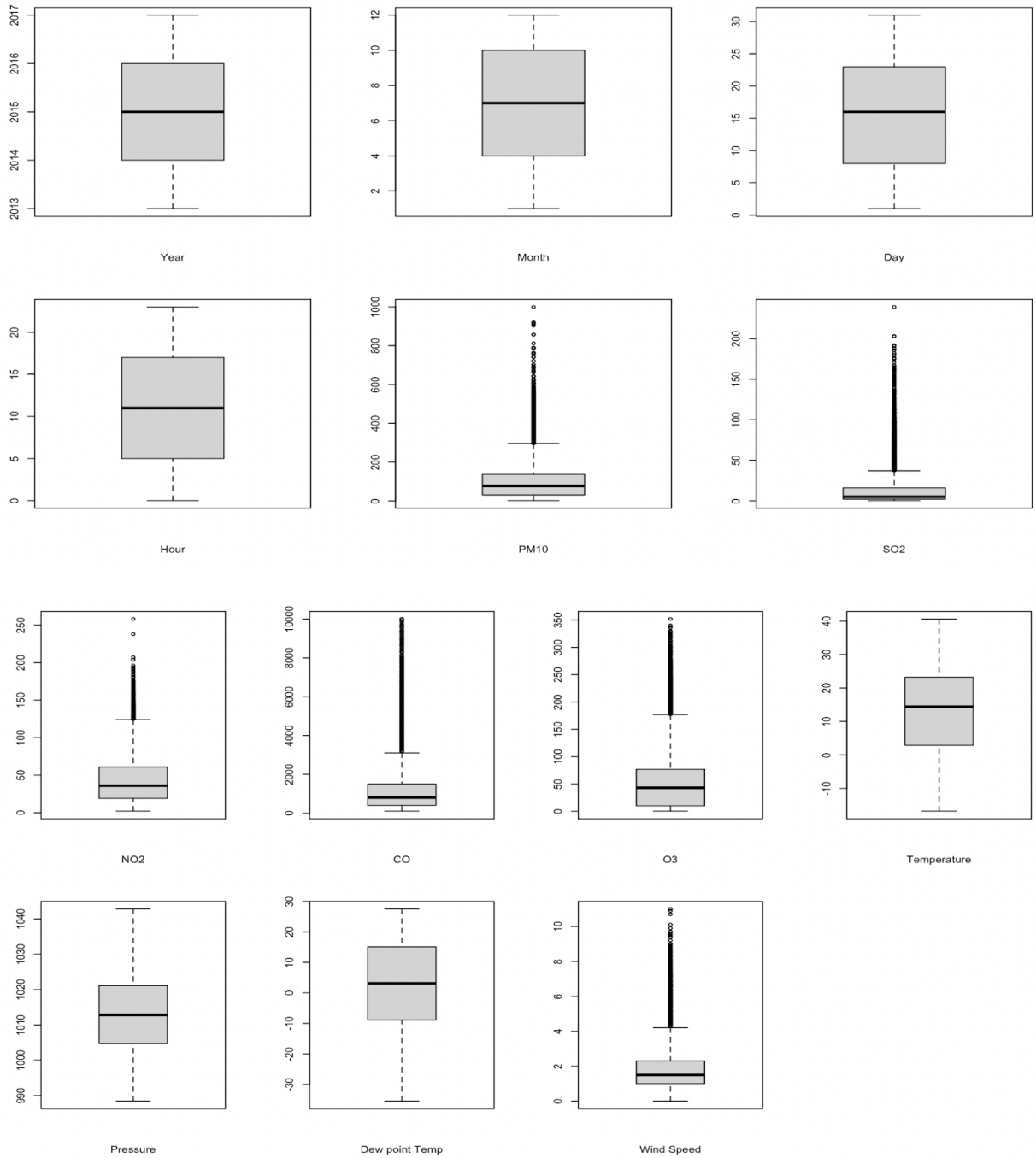


Figure A2.

Histograms of each of the predictor variables before transformations.

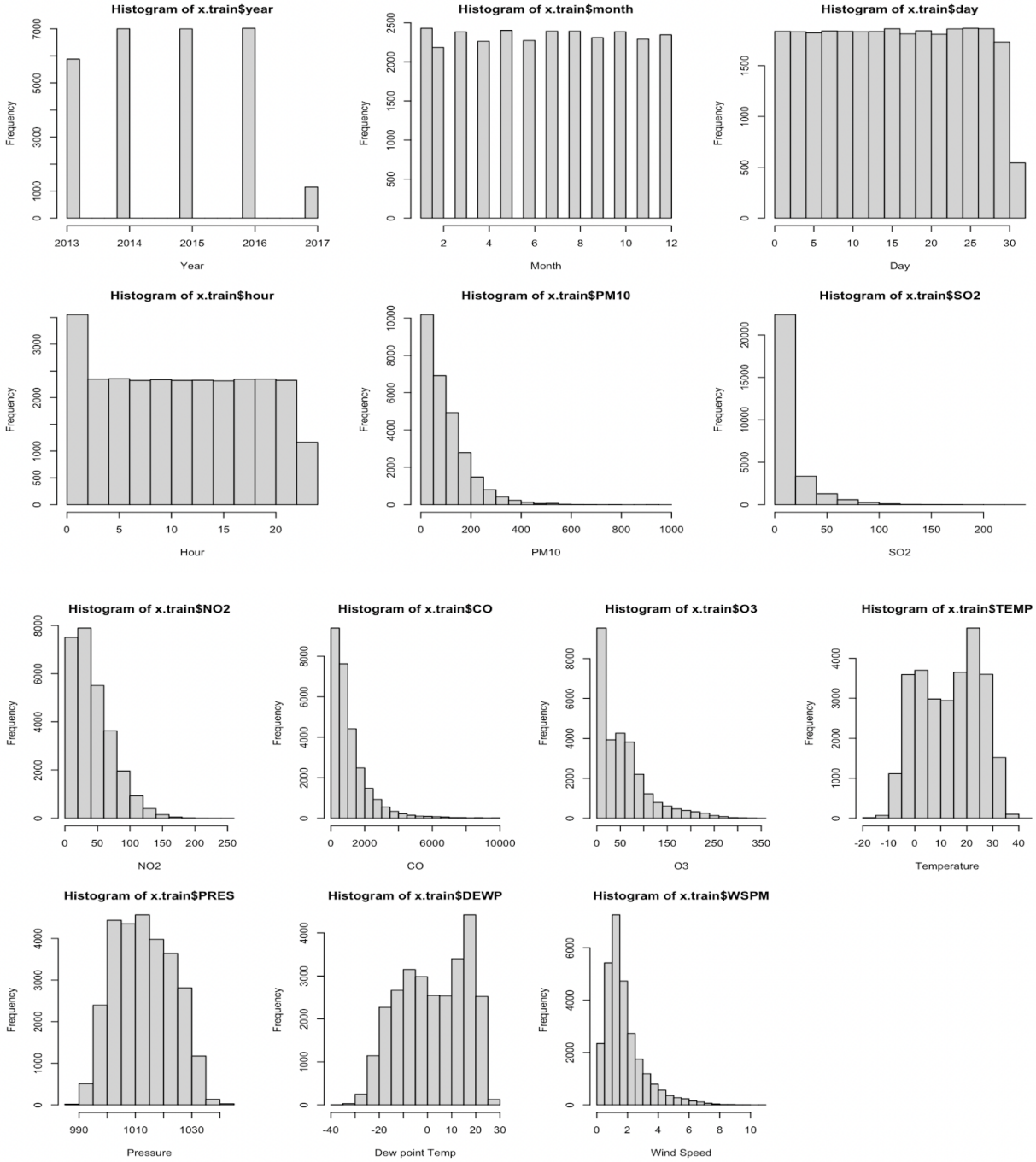


Figure A3.

Boxplots of each of the predictor variables after transformations.

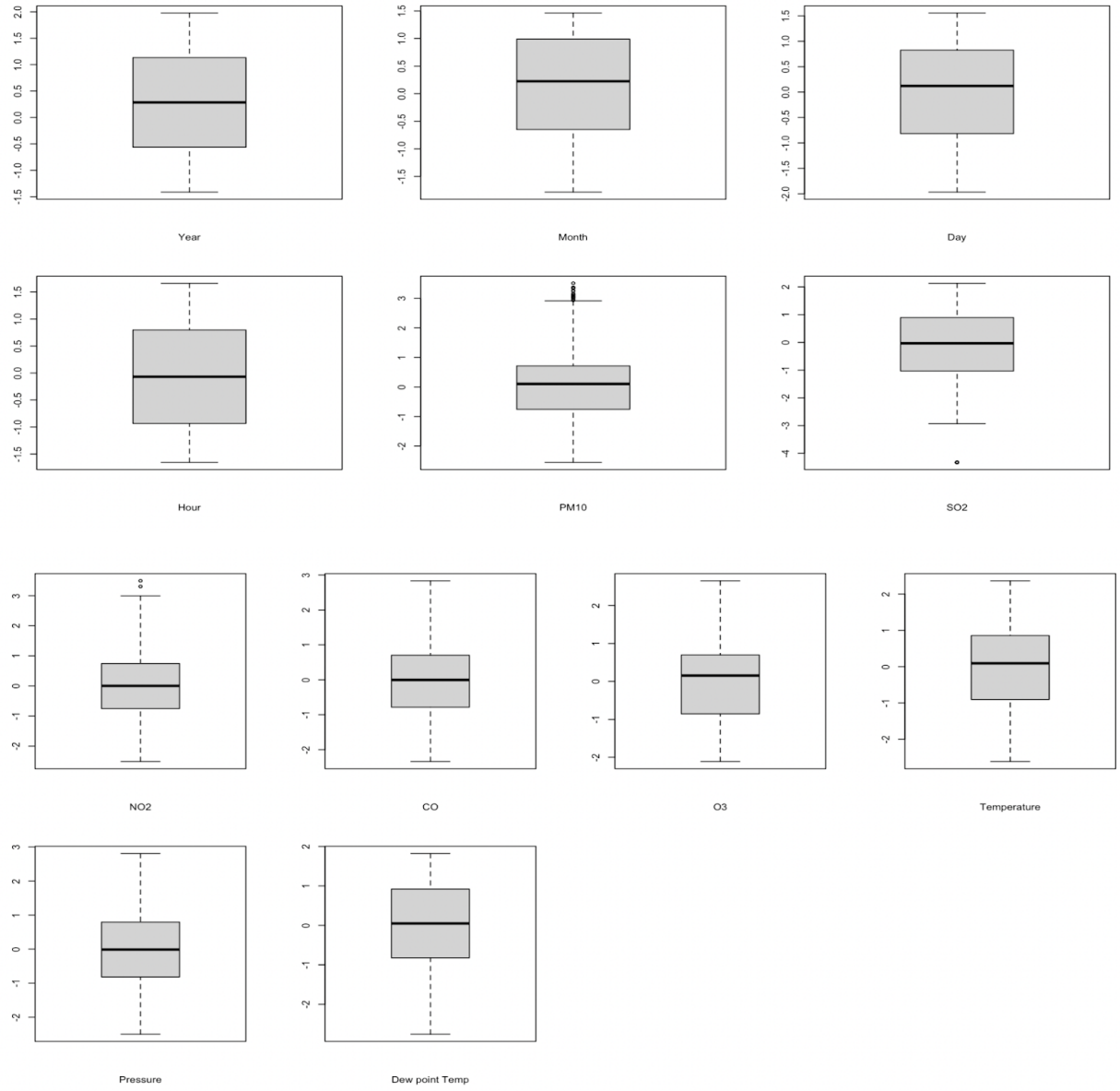


Figure A4.

Histograms of each of the predictor variables after transformations.

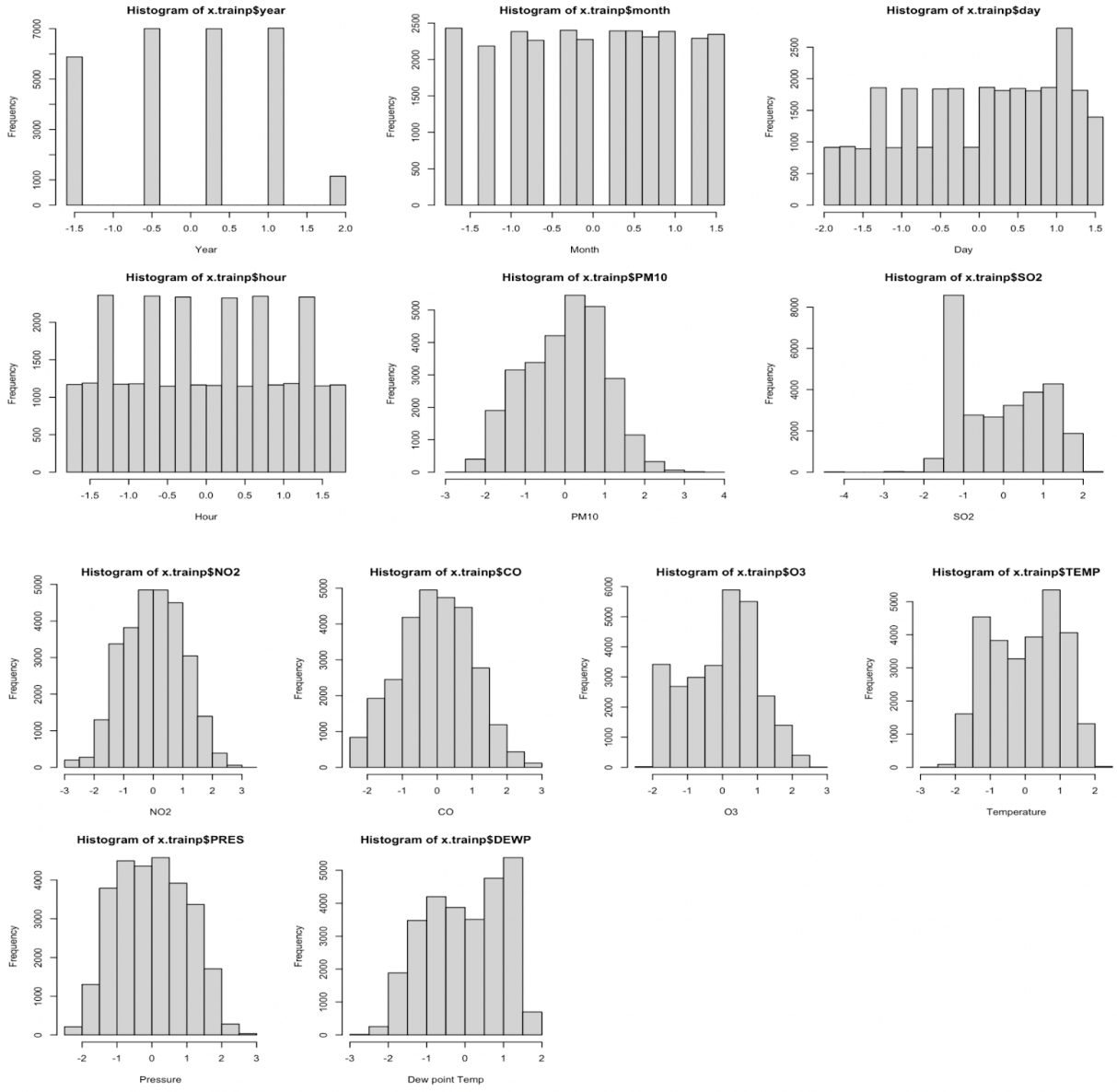
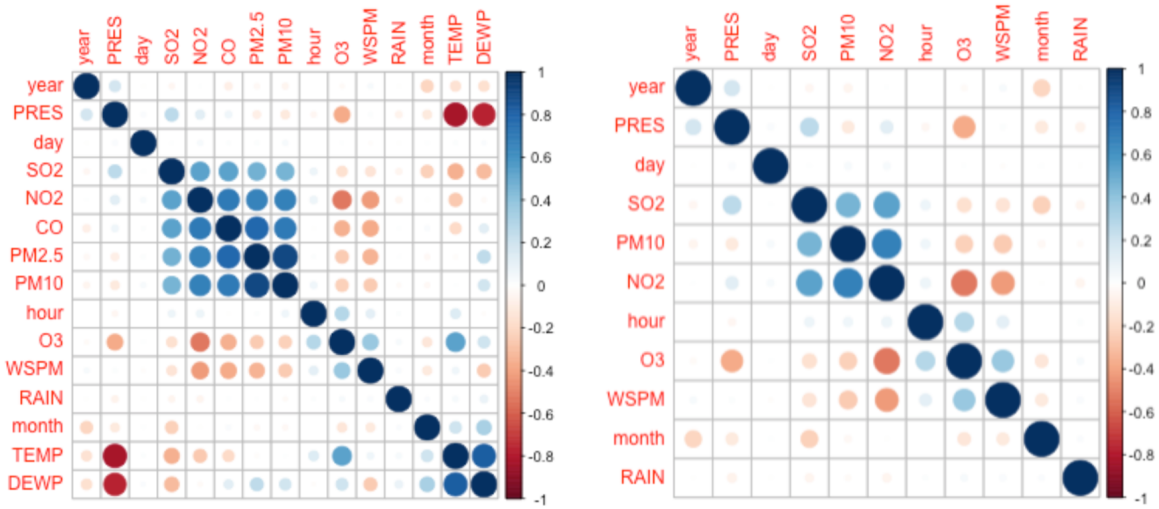


Figure A5.

Correlation heatmaps of before and after removing correlated variables.



Note. The correlation heatmap on the left was produced before the "box-cox", "centering", and "scaling" transformations. The correlation heatmap on the right was produced after the transformations.

Appendix B

Group1: Sean Torres and Anusia Edward

```
#Importing Dataset
air <- read.csv("~/Desktop/Shunyi.csv")
```

Data Pre-Processing

```
sum(is.na(air$PM2.5))
```

```
## [1] 913
```

```
# won't let me split because of the NAs in the outcome var
# filling outcome var
air.o <- kNN(air, variable = c("PM2.5"))
air.o <- subset(air.o, select = year:WSPM)
x <- subset(air.o, select = -PM2.5)
y <- subset(air.o, select = PM2.5)
# splitting dataset to ensure no further data leakage
set.seed(1)
trainset <- createDataPartition(air.o$PM2.5, p = 0.8, list = FALSE)
x.train <- x[trainset, ]
y.train <- y[trainset, ]
x.test <- x[-trainset, ]
y.test <- y[-trainset, ]
y.train1 <- as.data.frame(y.train)
y.test1 <- as.data.frame(y.test)
#imputing missing values using KNN
sum(is.na(x.train))
```

```
## [1] 5782
```

```
x.train <- kNN(x.train)
x.train <- subset(x.train, select = year:WSPM)
sum(is.na(x.test))
```

```
## [1] 1345
```

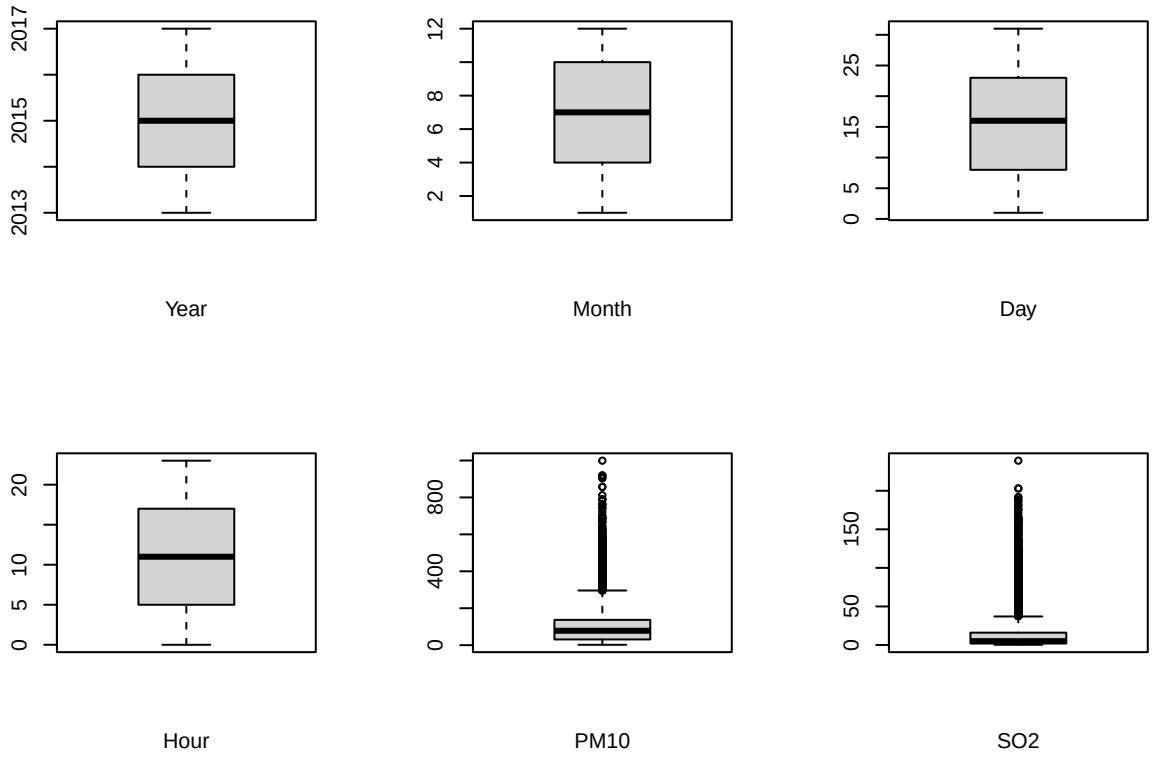
```
x.test <- kNN(x.test)
x.test <- subset(x.test, select = year:WSPM)
sum(is.na(y.train))
```

```
## [1] 0
```

```
sum(is.na(y.test))
```

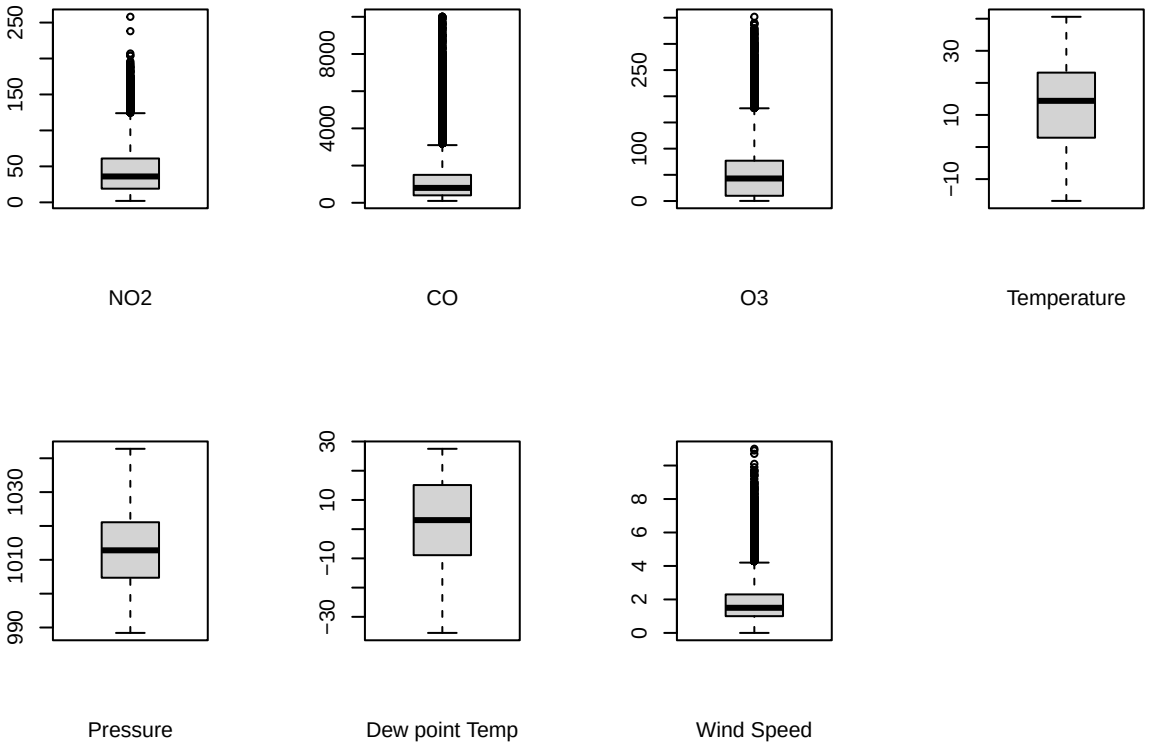
```
## [1] 0
```

```
# near Zero Variance removal
nZV.x <- nearZeroVar(x.train)
x.train <- x.train[, -nZV.x]
x.test <- x.test[, -nZV.x]
# visualizing outliers
par(mfrow = c(2,3))
boxplot(x.train$year,xlab = "Year")
boxplot(x.train$month, xlab = "Month")
boxplot(x.train$day, xlab = "Day")
boxplot(x.train$hour, xlab = "Hour")
boxplot(x.train$PM10, xlab = "PM10")
boxplot(x.train$SO2, xlab = "SO2")
```

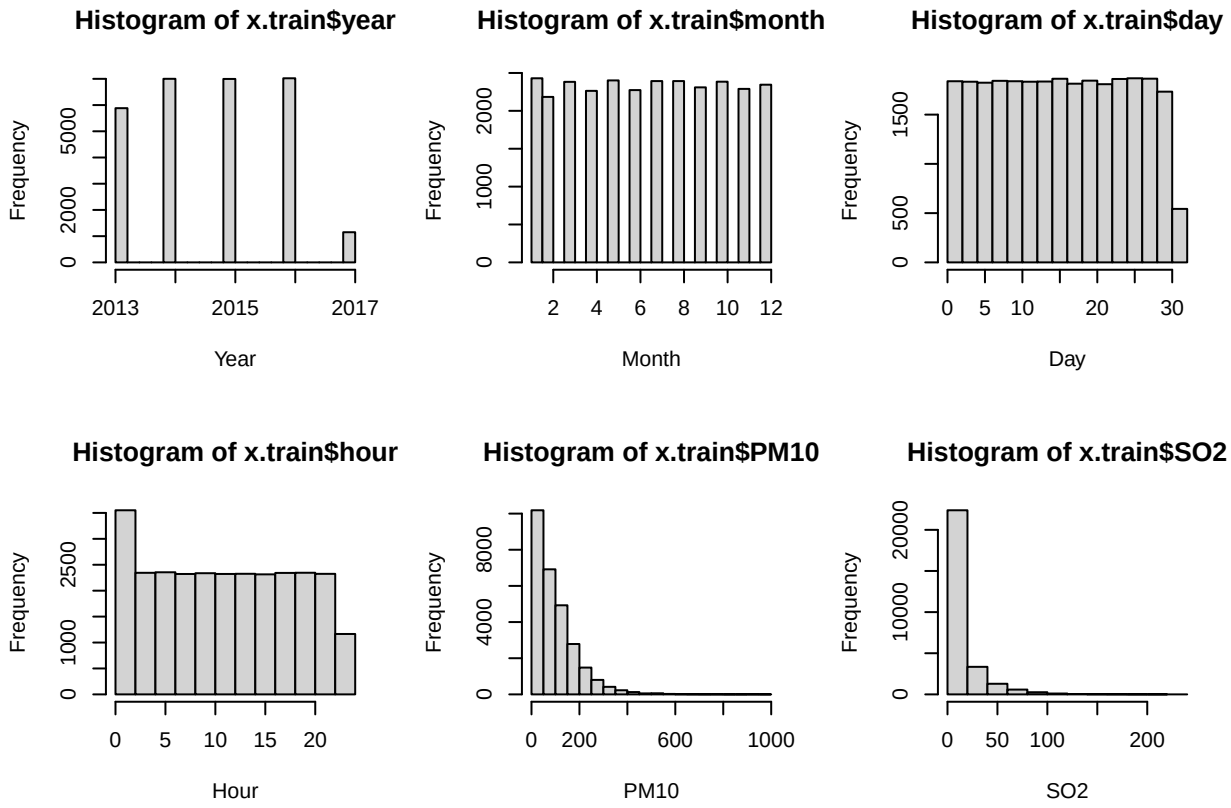



```
par(mfrow = c(2,4))
boxplot(x.train$NO2, xlab = "NO2")
boxplot(x.train$CO, xlab = "CO")
boxplot(x.train$O3, xlab = "O3")
boxplot(x.train$TEMP, xlab = "Temperature")
boxplot(x.train$PRES, xlab = "Pressure")
boxplot(x.train$DEWP, xlab = "Dew point Temp")
```

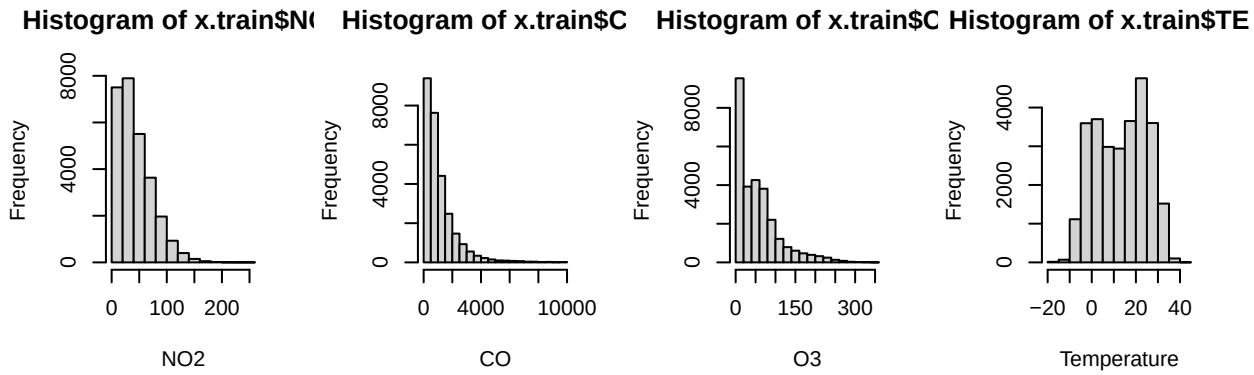
```
boxplot(x.train$WSPM, xlab = "Wind Speed")
# visualizing distributions
par(mfrow = c(2,3))
```

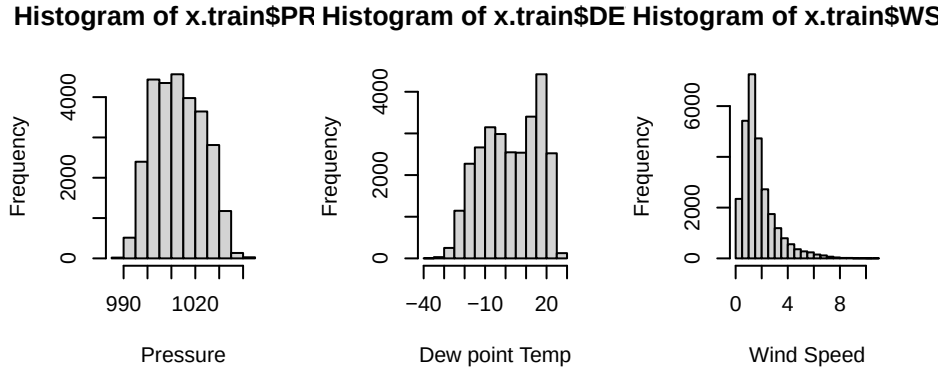



```
hist(x.train$year,xlab = "Year")
hist(x.train$month, xlab = "Month")
hist(x.train$day, xlab = "Day")
hist(x.train$hour, xlab = "Hour")
hist(x.train$PM10, xlab = "PM10")
hist(x.train$SO2, xlab = "SO2")
```

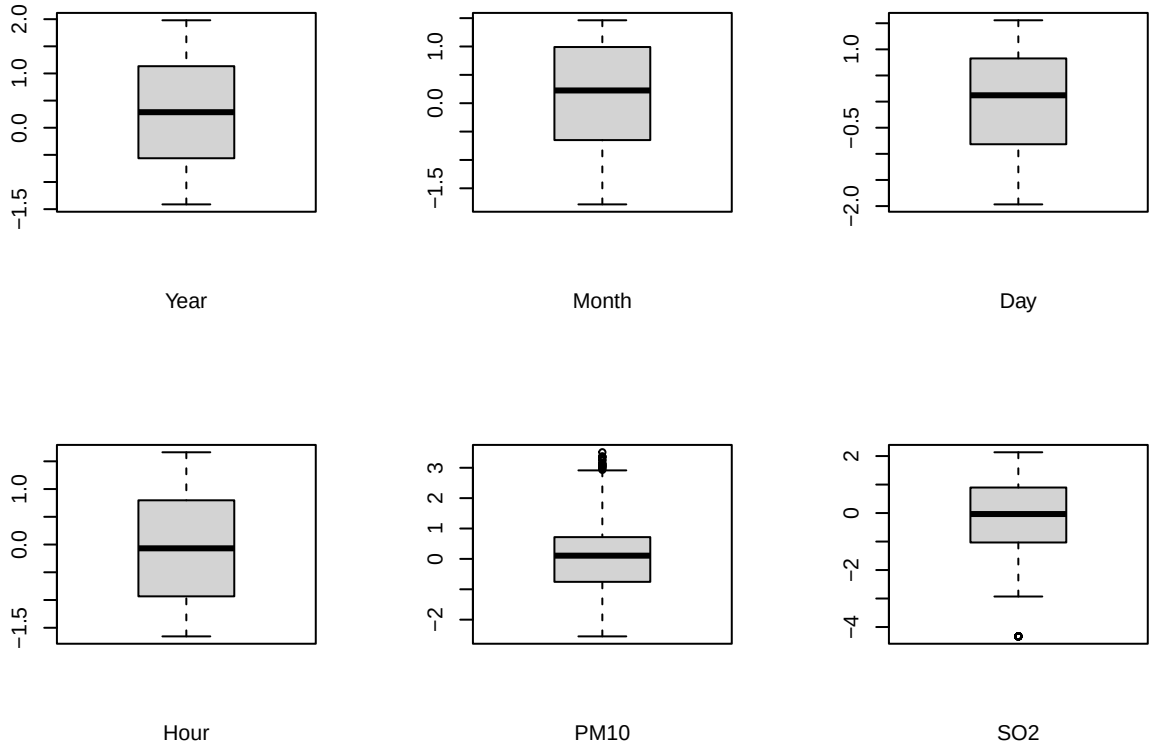



```
par(mfrow = c(2,4))
hist(x.train$NO2, xlab = "NO2")
hist(x.train$CO, xlab = "CO")
hist(x.train$O3, xlab = "O3")
hist(x.train$TEMP, xlab = "Temperature")
hist(x.train$PRES, xlab = "Pressure")
hist(x.train$DEWP, xlab = "Dew point Temp")
hist(x.train$WSPM, xlab = "Wind Speed")
# box-cox, center, scaling
trans <- preProcess(x.train,
                    method = c("BoxCox", "center", "scale"))
x.trainp <- predict(trans, x.train)
trans1 <- preProcess(y.train1,
                    method = c("BoxCox", "center", "scale"))
y.trainp <- predict(trans1, y.train1)
trans2 <- preProcess(x.test,
                    method = c("BoxCox", "center", "scale"))
x.testp <- predict(trans2, x.test)
trans3 <- preProcess(y.test1,
                    method = c("BoxCox", "center", "scale"))
y.testp <- predict(trans3, y.test1)
# visualizing outliers after transformations
par(mfrow = c(2,3))
```

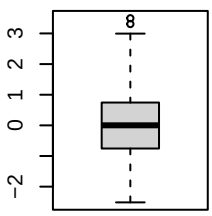




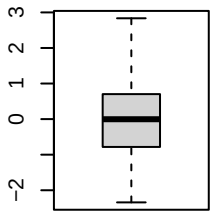

```
boxplot(x.trainp$year,xlab = "Year")
boxplot(x.trainp$month, xlab = "Month")
boxplot(x.trainp$day, xlab = "Day")
boxplot(x.trainp$hour, xlab = "Hour")
boxplot(x.trainp$PM10, xlab = "PM10")
boxplot(x.trainp$SO2, xlab = "SO2")
```



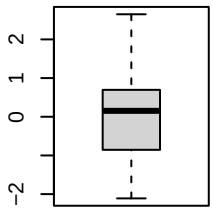

```
par(mfrow = c(2,4))
boxplot(x.trainp$NO2, xlab = "NO2")
boxplot(x.trainp$CO, xlab = "CO")
boxplot(x.trainp$O3, xlab = "O3")
boxplot(x.trainp$TEMP, xlab = "Temperature")
boxplot(x.trainp$PRES, xlab = "Pressure")
boxplot(x.trainp$DEWP, xlab = "Dew point Temp")
boxplot(x.trainp$WSPM, xlab = "Wind Speed")
# visualizing distribution after transformations
par(mfrow = c(2,3))
```



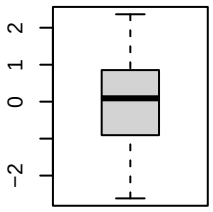
NO2



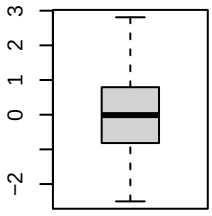
CO



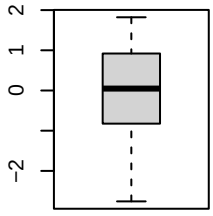
O3



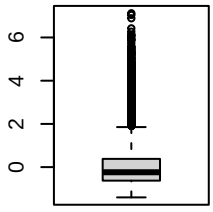
Temperature



Pressure

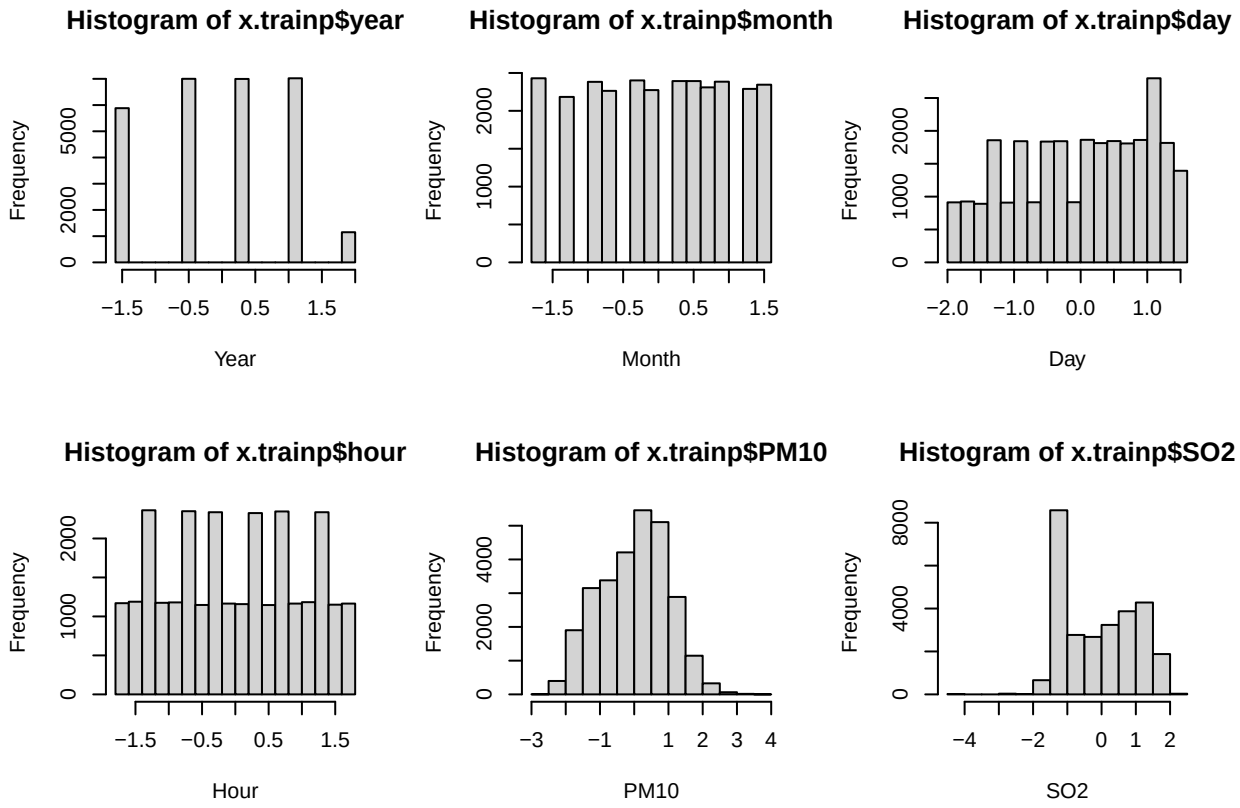


Dew point Temp

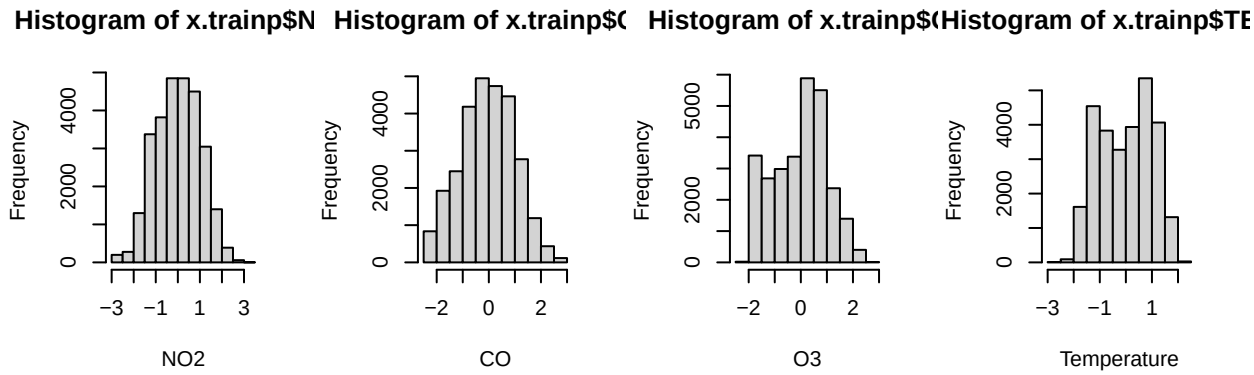


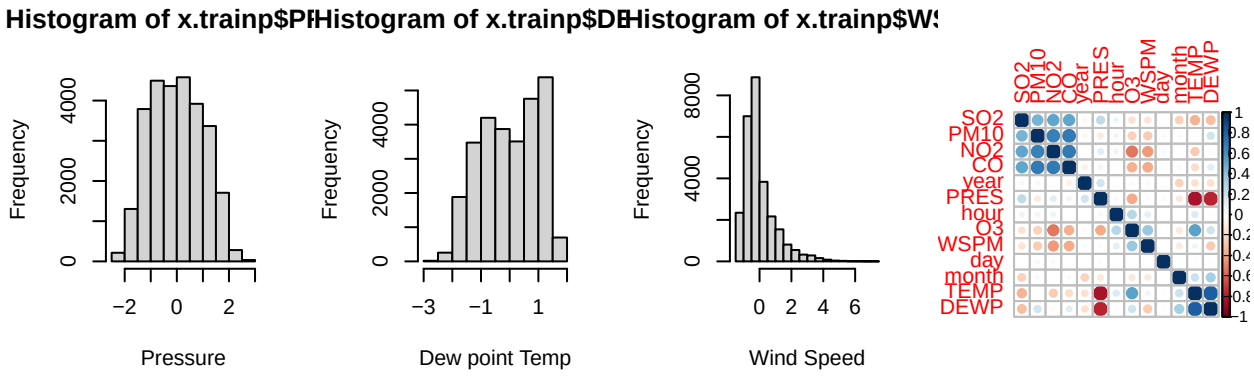
Wind Speed

```
hist(x.trainp$year,xlab = "Year")
hist(x.trainp$month, xlab = "Month")
hist(x.trainp$day, xlab = "Day")
hist(x.trainp$hour, xlab = "Hour")
hist(x.trainp$PM10, xlab = "PM10")
hist(x.trainp$SO2, xlab = "SO2")
```




```
par(mfrow = c(2,4))
hist(x.trainp$NO2, xlab = "NO2")
hist(x.trainp$CO, xlab = "CO")
hist(x.trainp$O3, xlab = "O3")
hist(x.trainp$TEMP, xlab = "Temperature")
hist(x.trainp$PRES, xlab = "Pressure")
hist(x.trainp$DEWP, xlab = "Dew point Temp")
hist(x.trainp$WSPM, xlab = "Wind Speed")
# visualizing correlation
x.corr <- cor(x.trainp)
corrplot(x.corr, order = "hclust")
```



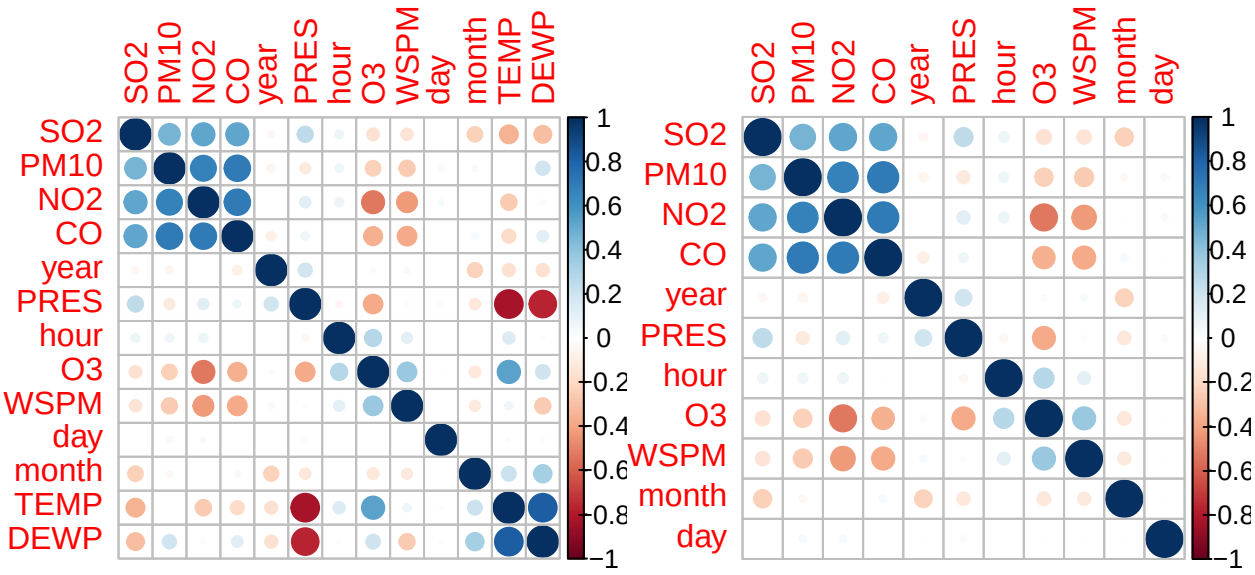



```
hCorr <- findCorrelation(x.corr, cutoff = 0.75, exact = TRUE)
x.trainpc <- x.trainp[, -hCorr]
x.testpc <- x.testp[, -hCorr]
x.corrCheck <- cor(x.trainpc)
x.corrCheck
```

```
##                year        month          day          hour        PM10
## year   1.0000000000 -0.220995074 -0.004555605 -0.0003806929 -0.05659725
## month -0.2209950737  1.000000000  0.004912318  0.0004323410 -0.03266403
## day   -0.0045556049  0.004912318  1.000000000  0.0015220176  0.03788677
## hour  -0.0003806929  0.000432341  0.001522018  1.0000000000  0.06790871
## PM10  -0.0565972501 -0.032664030  0.037886774  0.0679087075  1.00000000
## SO2   -0.0455255923 -0.233431501  0.001768788  0.0724217371  0.46332563
## NO2    0.0005176724 -0.007179976  0.031387206  0.0778285408  0.67247614
## CO    -0.0874638065  0.037719054  0.007406187 -0.0045958308  0.70132058
## O3    -0.0216861358 -0.129249680  0.007955603  0.2874547673 -0.23517020
## PRES   0.1909912799 -0.122466226  0.026467936 -0.0421577507 -0.11540840
## WSPM   0.0310687386 -0.117742917 -0.005800847  0.1139853998 -0.25470725
##                SO2           NO2           CO           O3        PRES
## year  -0.045525592  0.0005176724 -0.087463807 -0.021686136  0.19099128
## month -0.233431501 -0.0071799758  0.037719054 -0.129249680 -0.12246623
## day    0.001768788  0.0313872056  0.007406187  0.007955603  0.02646794
## hour   0.072421737  0.0778285408 -0.004595831  0.287454767 -0.04215775
## PM10   0.463325633  0.6724761422  0.701320580 -0.235170198 -0.11540840
## SO2    1.000000000  0.5292395310  0.526082806 -0.154902642  0.25283278
## NO2    0.529239531  1.0000000000  0.701677378 -0.524489511  0.12005221
```

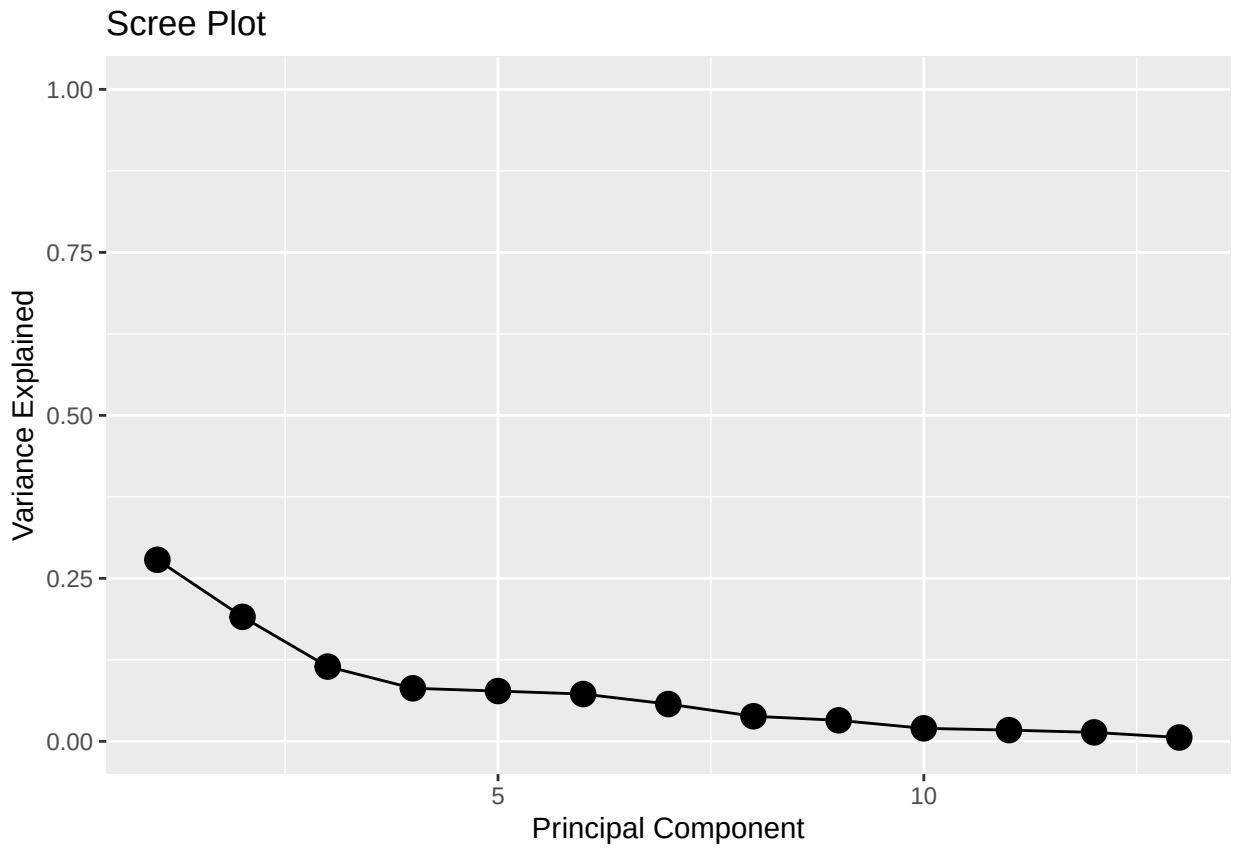
```
## CO     0.526082806  0.7016773783  1.000000000 -0.354901956  0.06123205
## O3    -0.154902642 -0.5244895112 -0.354901956  1.000000000 -0.37528681
## PRES   0.252832784  0.1200522100  0.061232054 -0.375286809  1.00000000
## WSPM  -0.143645492 -0.4282418879 -0.378932137  0.378000593  0.01706782
##                WSPM
## year   0.031068739
## month -0.117742917
## day   -0.005800847
## hour   0.113985400
## PM10  -0.254707254
## SO2   -0.143645492
## NO2   -0.428241888
## CO    -0.378932137
## O3     0.378000593
## PRES   0.017067816
## WSPM   1.000000000
```

```
par(mfrow = c(1,2))
corrplot(x.corr, order = "hclust")
corrplot(x.corrCheck, order = "hclust")
```



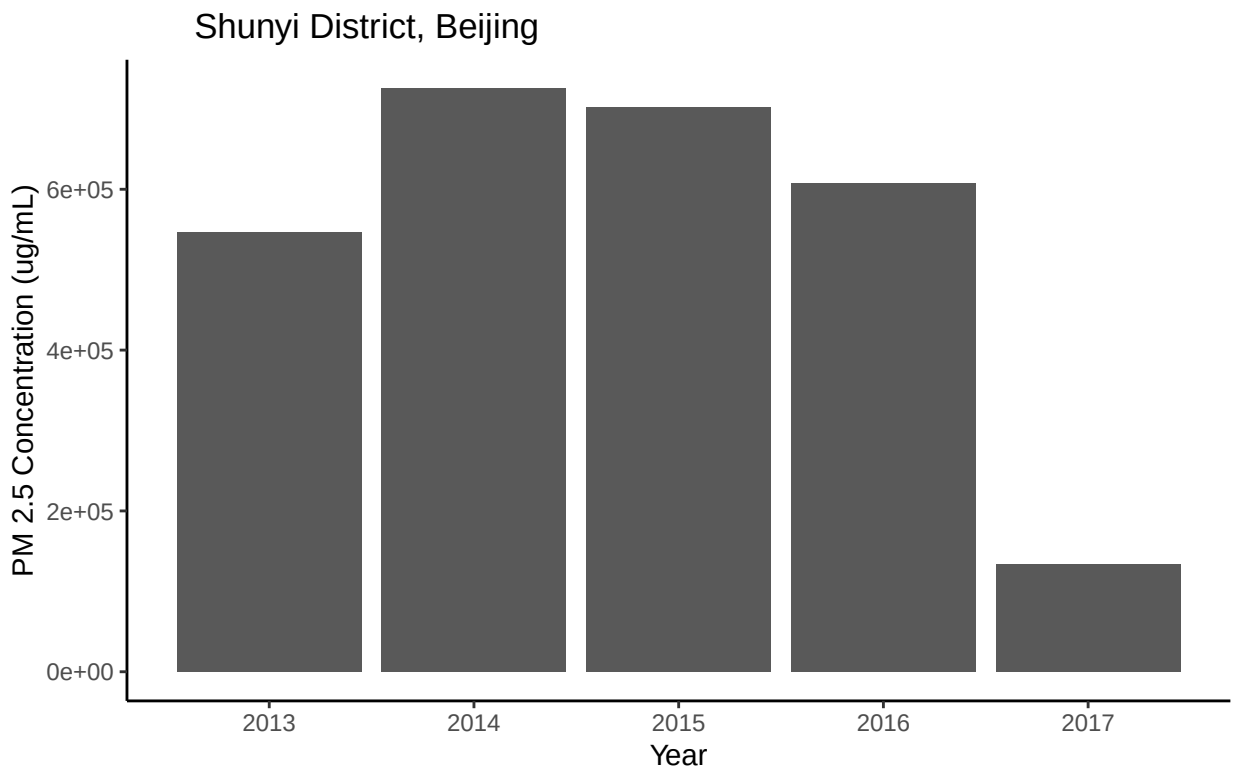

```
# PCA
pca.x <- prcomp(x.train, center = TRUE, scale. = TRUE)
variance = pca.x$sdev^2 / sum(pca.x$sdev^2)
# variance
```

```
qplot(c(1:13), variance) +
  geom_line() +
  geom_point(size=4)+
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("Scree Plot") +
  ylim(0, 1)
```

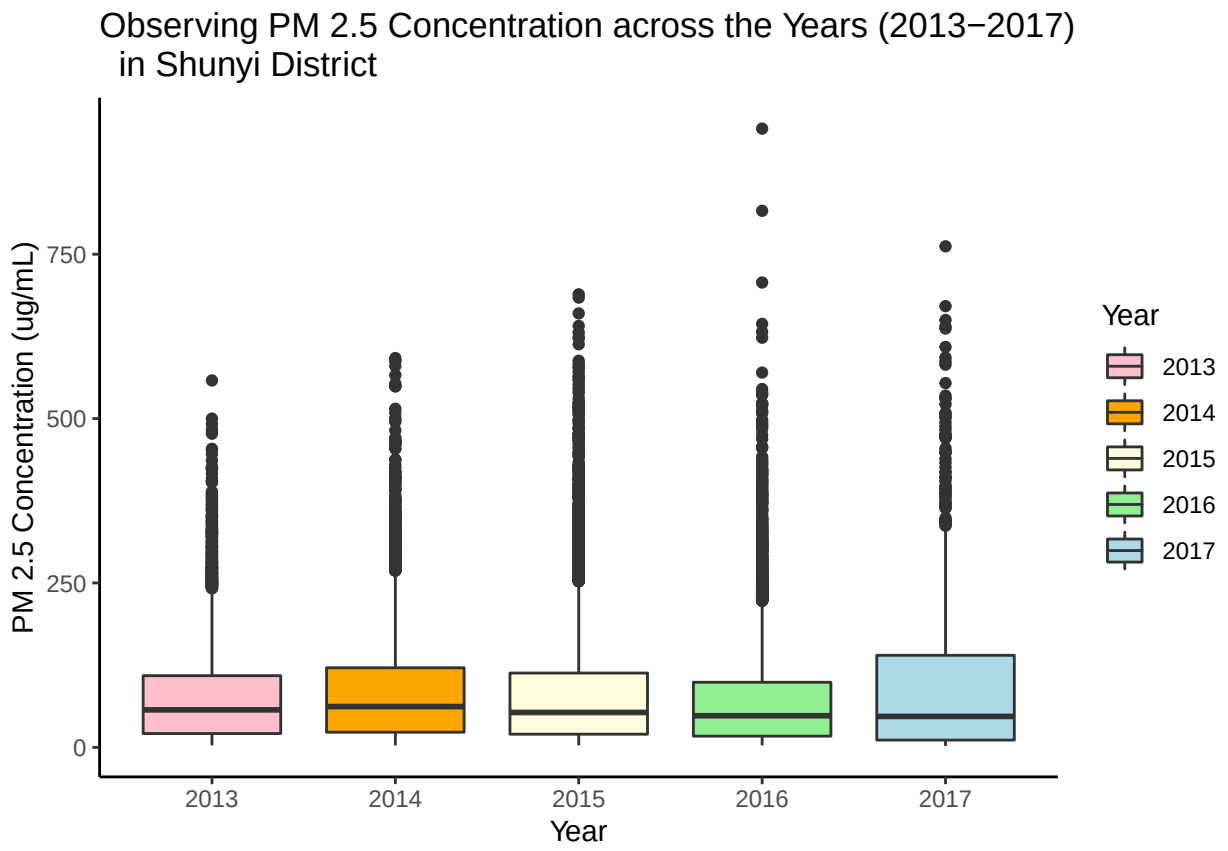


Exploratory Data Analysis

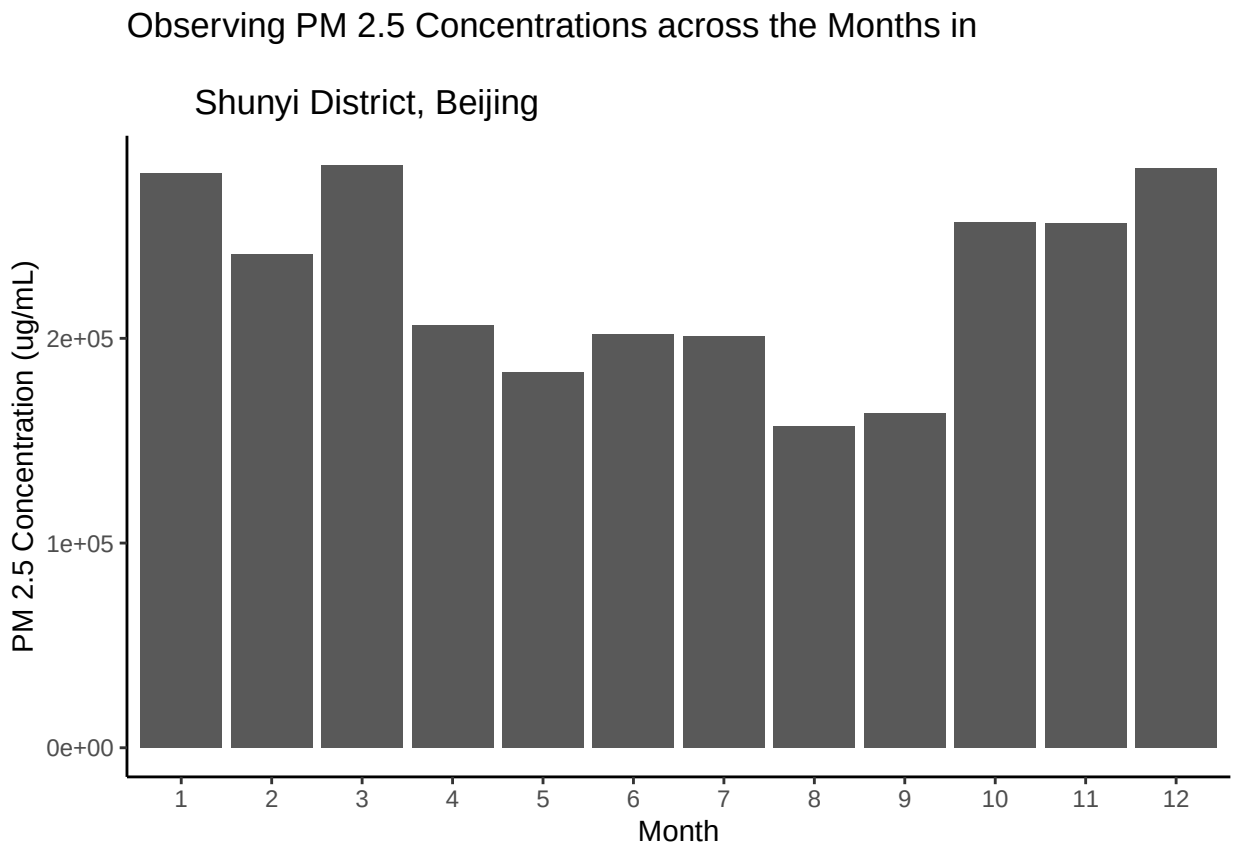
```
# these visualizations are observing data before it was pre-processed
# Observing PM 2.5 Concentrations across the Years (2013-2017)
ggplot(data = air, aes(x=year,y=PM2.5)) +
geom_bar(stat = "identity") +
theme_minimal() +
  labs(x = "Year", y = "PM 2.5 Concentration (ug/mL)",
       title =
         "Observing PM 2.5 Concentrations across the Years (2013-2017) in \n
       Shunyi District, Beijing", hjust = 0.5) +
  theme_classic()
```




```
# Boxplot PM 2.5 Concentrations across the Years (2013-2017)
ggplot(data = air, aes(x=year,y=PM2.5)) +
geom_boxplot(aes(fill=factor(year), fill = year)) +
theme_minimal() +
  scale_fill_manual(name = "Year", labels = c("2013",
                                              "2014",
                                              "2015",
                                              "2016",
                                              "2017"),
          values = c("pink","orange", "light yellow","light green",
                     "light blue")) +
  labs(x = "Year", y = "PM 2.5 Concentration (ug/mL)", title =
  "Observing PM 2.5 Concentration across the Years (2013-2017)
  in Shunyi District", adj = 0.5) +
  theme_classic()
```

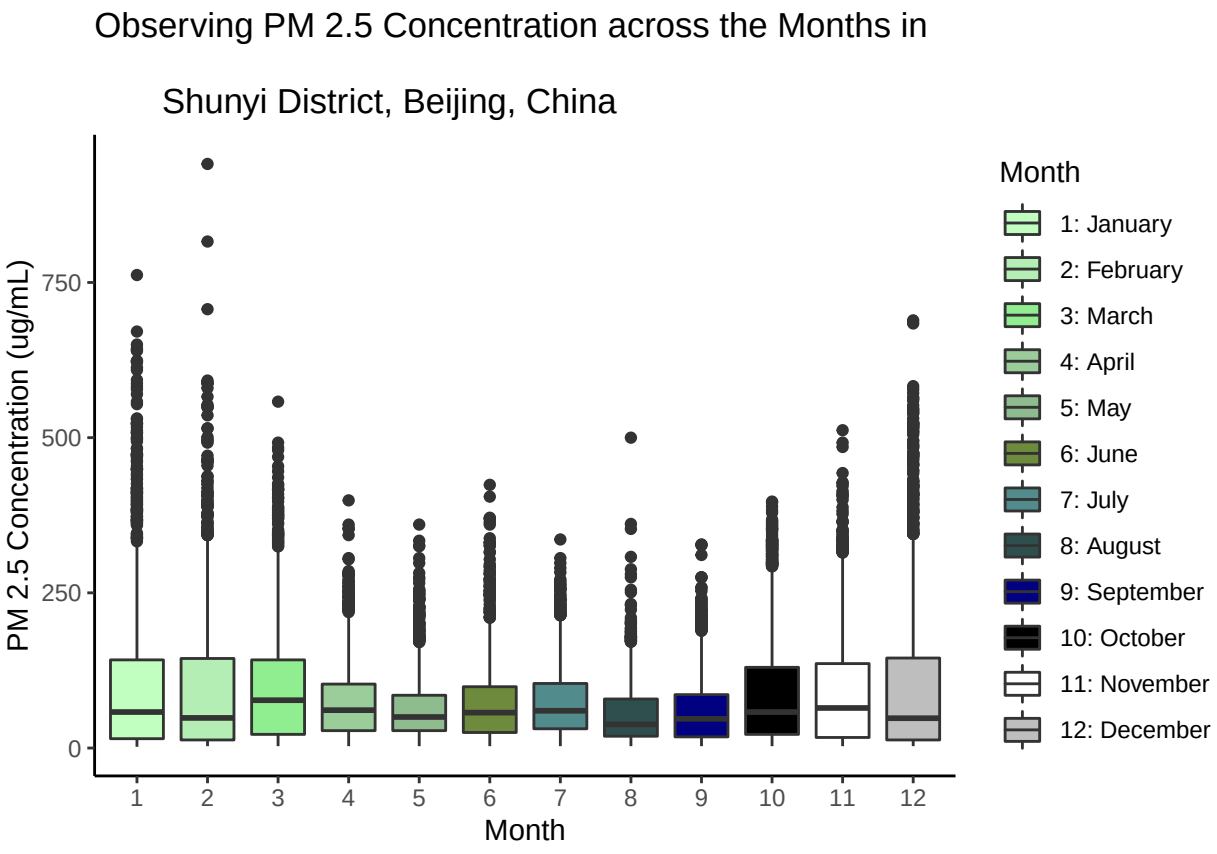



```
# Observing PM 2.5 Concentrations across the Months
ggplot(data = air, aes(x=month,y=PM2.5)) +
geom_bar(stat = "identity") +
theme_minimal() +
  labs(x = "Month", y = "PM 2.5 Concentration (ug/mL)",
       title = "Observing PM 2.5 Concentrations across the Months in \n
       Shunyi District, Beijing",
       adj = 0.5) +
   scale_x_discrete(name = "Month",
                    limits = c("1", "2", "3", "4", "5", "6","7", "8", "9", "10",
                               "11", "12")) +
  theme_classic()
```




```
# Boxplot of PM2.5 Concentrations per month
new_data = cbind(x.train, y.train)
ggplot(data = new_data, aes(x=month,y=y.train)) +
geom_boxplot(aes(fill=factor(month), fill = month)) +
theme_minimal() +
scale_color_manual(name = "Month", labels = c("1: January",
                                              "2: February",
                                              "3: March",
                                              "4: April",
                                          "5: May",
                                          "6: June",
                                          "7: July",
                                          "8: August",
                                          "9: September",
                                          "10: October",
                                         "11: November",
                                         "12: December"),
                   values = c("darkseagreen1","darkseagreen2", "light green",
                              "darkseagreen3",
                              "darkseagreen",
                              "darkolivegreen4",
                              "darkslategray4", "darkslategray", "navy",
                              "black", "white", "grey")) +
  scale_fill_manual(name = "Month", labels = c("1: January",
                                               "2: February",
                                               "3: March",
```

```
                                                "4: April",
                                              "5: May",
                                              "6: June",
                                              "7: July",
                                              "8: August",
                                              "9: September",
                                              "10: October",
                                             "11: November",
                                             "12: December"),
                    values = c("darkseagreen1","darkseagreen2", "light green",
                               "darkseagreen3",
                               "darkseagreen",
                               "darkolivegreen4",
                               "darkslategray4", "darkslategray", "navy",
                               "black", "white", "grey")) +
scale_x_discrete(name = "Month",
                    limits = c("1", "2", "3", "4", "5", "6","7", "8", "9", "10",
                               "11", "12")) +
labs(x = "Month", y = "PM 2.5 Concentration (ug/mL)",
     title =
       "Observing PM 2.5 Concentration across the Months in \n
     Shunyi District, Beijing, China",
     adj = 0.5) +
theme_classic()
```



Preliminary Models, Hyperparameter Tuning, and Model Evaluations

```
# OLS
# Using as a base model
set.seed(100)
indx <- createFolds(y.train, returnTrain = TRUE)
ctrl <- trainControl(method = "cv", index = indx)
pcrTune2 <- train(x = x.trainpc, y = y.train,
                  method = "lm",trControl = ctrl)
pcrTune2
```

```
## Linear Regression
##
## 28053 samples
##    11 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 25247, 25248, 25247, 25248, 25247, 25248, ...
## Resampling results:
##
##   RMSE     Rsquared   MAE
##   45.3015  0.6951456  31.54367
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
summary(pcrTune2)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -191.63  -27.21   -7.13   19.06  701.48
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  80.0284     0.2706 295.792  < 2e-16 ***
## year          2.6996     0.2863   9.428  < 2e-16 ***
## month        -1.8366     0.2968  -6.187 6.20e-10 ***
## day          -2.1772     0.2716  -8.017 1.12e-15 ***
## hour         -2.2280     0.2998  -7.431 1.11e-13 ***
## PM10         55.0953     0.4327 127.332  < 2e-16 ***
## SO2          -4.9880     0.3710 -13.444  < 2e-16 ***
## NO2          -4.0910     0.5030  -8.133 4.37e-16 ***
## CO           24.7482     0.4461  55.477  < 2e-16 ***
## O3            3.9193     0.4011   9.770  < 2e-16 ***
## PRES          8.1566     0.3370  24.206  < 2e-16 ***
## WSPM         -2.4262     0.3163  -7.671 1.76e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 45.32 on 28041 degrees of freedom
## Multiple R-squared:  0.695,  Adjusted R-squared:  0.6948
## F-statistic:  5808 on 11 and 28041 DF,  p-value: < 2.2e-16
```

```
# testResults
rfImp_OLS <- varImp(pcrTune2, scale = T)
rfImp_OLS
```

```
## lm variable importance
##
##       Overall
## PM10  100.000
## CO     40.686
## PRES   14.874
## SO2     5.990
## O3      2.957
## year    2.675
## NO2     1.606
## day     1.510
## WSPM    1.225
## hour    1.026
## month   0.000
```

```
fp_predict <- predict(pcrTune2, x.testpc)
postResample(fp_predict, y.test)
```

```
##      RMSE  Rsquared       MAE
## 43.249089  0.699256 30.864738
```

```
# PLS
set.seed(100)
indx <- createFolds(y.train, returnTrain = TRUE)
ctrl <- trainControl(method = "cv", index = indx)
pcrTune3 <- train(x = x.train, y = y.train,
                  method = "pls",
                  preProcess=c("center","scale"),
                  tuneGrid = expand.grid(ncomp = 1:14),
                  trControl = ctrl)
pcrTune3
```

```
## Partial Least Squares
##
## 28053 samples
##    13 predictor
##
## Pre-processing: centered (13), scaled (13)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 25247, 25248, 25247, 25248, 25247, 25248, ...
## Resampling results across tuning parameters:
##
##   ncomp  RMSE      Rsquared   MAE
```

```
##    1     43.96552  0.7124556  30.58076
##    2     38.29961  0.7817798  26.32104
##    3     34.12245  0.8267487  21.20946
##    4     32.99449  0.8380107  20.06535
##    5     32.50372  0.8427737  19.53363
##    6     32.37996  0.8439909  19.58367
##    7     32.32736  0.8445200  19.37568
##    8     32.28204  0.8449567  19.58626
##    9     32.27998  0.8449769  19.58984
##   10     32.27954  0.8449811  19.58751
##   11     32.27959  0.8449806  19.58824
##   12     32.27955  0.8449811  19.58853
##   13     32.27957  0.8449809  19.58874
##   14     32.27957  0.8449809  19.58874
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was ncomp = 10.
```

```
summary(pcrTune3)
```

```
## Data:    X dimension: 28053 13
##  Y dimension: 28053 1
## Fit method: oscorespls
## Number of components considered: 10
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X           24.13    44.07    52.89    61.34    66.79    71.91    76.93
## .outcome    71.27    78.19    82.70    83.83    84.31    84.43    84.48
##           8 comps  9 comps  10 comps
## X           78.15    84.91     87.18
## .outcome    84.52    84.52     84.52
```

```
fp_predict1 <- predict(pcrTune3, x.test)

postResample(fp_predict, y.test)
```

```
##      RMSE  Rsquared       MAE
## 43.249089  0.699256 30.864738
```

```
rfImp_PLS <- varImp(pcrTune3, scale = T)
rfImp_PLS
```

```
## pls variable importance
##
##       Overall
## PM10  100.000
## CO     81.674
## NO2    68.047
## SO2    50.017
## WSPM   30.027
## O3     16.971
## TEMP   15.615
```

```
## DEWP   13.557
## PRES    3.205
## month   3.140
## hour    1.995
## year    1.151
## day     0.000
```

```
# Random Forest
rfmodel <- randomForest(x = x.train, y = y.train,importance=TRUE,ntrees=500)

getRMSE <- function(x,y) {
  sqrt(sum((x-y)^2)/length(x))
}

testResults <- data.frame(obs = y.test,
                          rfmodel = predict(rfmodel, x.test))

getRMSE(testResults$obs, testResults$rfmodel)
```

```
## [1] 21.33502
```

```
fp_predict2 <- predict(rfmodel , x.testp)
# fp_predict2 (commented out because there were too many predictions)
summary(fp_predict2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24.56   29.02   35.00   33.04   35.52   39.55
```

```
postResample(fp_predict2, y.test)
```

```
##         RMSE     Rsquared          MAE
## 9.121772e+01 6.878555e-05 5.896752e+01
```

```
rfImp_RF <- varImp(rfmodel, scale = T)
rfImp_RF
```

```
##         Overall
## year   39.32902
## month  26.88878
## day    68.77043
## hour   49.77910
## PM10  134.12958
## SO2    41.97516
## NO2    34.03051
## CO     61.72128
## O3     37.31964
## TEMP   41.96578
## PRES   39.57512
## DEWP   41.20180
## WSPM   34.20973
```

```r
# Elastic Net
enetGrid <- expand.grid(lambda = c(0, 0.01, .1),
                        fraction = seq(.05, 1, length = 20))
set.seed(100)
enetTune <- train(x = x.trainp, y = y.train,
                  method = "enet",
                  tuneGrid = enetGrid,
                  trControl = ctrl,
                  preProc = c("center", "scale"))
enetTune
```

```
## Elasticnet
##
## 28053 samples
##    13 predictor
##
## Pre-processing: centered (13), scaled (13)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 25247, 25248, 25247, 25248, 25247, 25248, ...
## Resampling results across tuning parameters:
##
##   lambda  fraction  RMSE      Rsquared   MAE
##   0.00    0.05      76.69624  0.6442943  55.56063
##   0.00    0.10      71.62933  0.6442943  51.12443
##   0.00    0.15      66.85991  0.6442943  46.84017
##   0.00    0.20      62.45627  0.6442943  42.72756
##   0.00    0.25      58.45161  0.6627185  38.92585
##   0.00    0.30      54.86191  0.6728596  35.49089
##   0.00    0.35      51.77449  0.6780405  33.04523
##   0.00    0.40      49.28420  0.6806879  31.67756
##   0.00    0.45      47.48534  0.6819712  31.22058
##   0.00    0.50      46.41344  0.6853695  31.23297
##   0.00    0.55      45.84108  0.6890681  31.41194
##   0.00    0.60      45.56278  0.6922093  31.40713
##   0.00    0.65      45.35071  0.6949265  31.32230
##   0.00    0.70      45.17407  0.6972292  31.22923
##   0.00    0.75      45.02111  0.6992095  31.15339
##   0.00    0.80      44.89670  0.7007998  31.10540
##   0.00    0.85      44.80235  0.7019902  31.08550
##   0.00    0.90      44.73611  0.7028189  31.09045
##   0.00    0.95      44.69741  0.7033019  31.11632
##   0.00    1.00      44.68455  0.7034688  31.16275
##   0.01    0.05      76.84702  0.6442943  55.69120
##   0.01    0.10      71.91566  0.6442943  51.37780
##   0.01    0.15      67.26138  0.6442943  47.20666
##   0.01    0.20      62.94578  0.6442943  43.19495
##   0.01    0.25      59.00899  0.6622618  39.47508
##   0.01    0.30      55.45352  0.6725819  36.04051
##   0.01    0.35      52.35663  0.6778734  33.45819
##   0.01    0.40      49.80427  0.6805915  31.90402
##   0.01    0.45      47.88388  0.6819215  31.25982
##   0.01    0.50      46.66686  0.6835158  31.26665
##   0.01    0.55      45.97784  0.6878727  31.36382
##   0.01    0.60      45.65257  0.6908838  31.49907
```

```
##   0.01    0.65      45.42778  0.6937940  31.40977
##   0.01    0.70      45.24464  0.6961925  31.32397
##   0.01    0.75      45.08759  0.6982443  31.24948
##   0.01    0.80      44.95463  0.6999680  31.19262
##   0.01    0.85      44.85013  0.7013123  31.16243
##   0.01    0.90      44.77282  0.7023052  31.15562
##   0.01    0.95      44.72064  0.7029802  31.16703
##   0.01    1.00      44.69370  0.7033439  31.20023
##   0.10    0.05      77.52474  0.6442943  56.27742
##   0.10    0.10      73.20986  0.6442943  52.51900
##   0.10    0.15      69.09193  0.6442943  48.86199
##   0.10    0.20      65.21091  0.6449154  45.32252
##   0.10    0.25      61.63186  0.6640972  42.03086
##   0.10    0.30      58.30457  0.6736883  38.87298
##   0.10    0.35      55.27383  0.6785310  35.95402
##   0.10    0.40      52.59111  0.6809628  33.67057
##   0.10    0.45      50.31232  0.6821044  32.17746
##   0.10    0.50      48.49461  0.6825292  31.40031
##   0.10    0.55      47.19136  0.6825468  31.22433
##   0.10    0.60      46.41736  0.6832965  31.47757
##   0.10    0.65      45.96037  0.6867363  31.67839
##   0.10    0.70      45.76135  0.6889784  31.97151
##   0.10    0.75      45.61883  0.6909958  31.98278
##   0.10    0.80      45.48748  0.6928652  31.94907
##   0.10    0.85      45.36863  0.6945581  31.91679
##   0.10    0.90      45.28298  0.6958064  31.90855
##   0.10    0.95      45.21964  0.6967780  31.92103
##   0.10    1.00      45.17449  0.6975326  31.94838
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were fraction = 1 and lambda = 0.
```

```
enet_predict <- predict(enetTune, x.testp)
# enet_predict (commented out because there were too many predictions)
summary(enet_predict)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -115.65   27.65   82.33   80.03  129.27  300.80
```

```
postResample(enet_predict, y.test)
```

```
##       RMSE   Rsquared        MAE
## 42.9263146  0.7038906 30.6430285
```

```
rfImp_EN <- varImp(enetTune, scale = T)
rfImp_EN
```

```
## loess r-squared variable importance
##
##         Overall
## PM10  1.000e+02
## CO    7.661e+01
```

```
## NO2   5.191e+01
## SO2   2.566e+01
## WSPM  1.086e+01
## O3    8.651e+00
## TEMP  2.483e+00
## DEWP  1.871e+00
## month 2.562e-01
## hour  3.357e-02
## PRES  1.489e-02
## year  5.978e-03
## day   0.000e+00
```

```
resamp <- resamples(list(OLS = pcrTune2, PLS = pcrTune3,Enet=enetTune))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: OLS, PLS, Enet
## Number of resamples: 10
##
## MAE
##          Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## OLS  30.60395 31.03170 31.55598 31.54367 31.88867 32.88581    0
## PLS  18.57821 19.43927 19.57927 19.58751 19.94144 20.36607    0
## Enet 30.25447 30.66429 31.24254 31.16275 31.52767 32.35359    0
##
## RMSE
##          Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## OLS  41.91173 44.64342 45.53493 45.30150 46.22377 48.12999    0
## PLS  28.73939 31.79096 32.83348 32.27954 33.09557 33.99087    0
## Enet 41.58058 44.19342 44.83082 44.68455 45.55638 47.42750    0
##
## Rsquared
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## OLS  0.6746818 0.6904134 0.6944912 0.6951456 0.7026644 0.7166186    0
## PLS  0.8179351 0.8355709 0.8462767 0.8449811 0.8557245 0.8662906    0
## Enet 0.6840361 0.7009725 0.7027909 0.7034688 0.7105767 0.7214876    0
```