Beijing PM2.5 Analysis

Alan Ji, Marcus Martinez, Paul Gao

November 2018

1 Introduction

Beijing, China is known across the world as a city plagued by severe air pollution. The most widely used measurement of air pollution is PM2.5, which is particulate matter in the air that is 2.5 microns or less in width, and it is an air pollutant that is of serious concern for people and our daily lives. The major environmental issue has been a constant battle in the last decade, and in this project, our purposes are to determine the major factors responsible for this pollution, discuss whether these factors have been targeted by recent initiatives, and predict future PM2.5 levels.

2 Main Questions

- What major atmospheric and temporal factors affect the overall PM2.5 concentration in Beijing?
- What can we conclude are the main source(s) of air pollution in this highly-populated city, and have the recent initiatives been in line with these conclusions?

3 The Data

To answer these questions, we sought a dataset that includes not only PM2.5 levels in a time-series format, but also various attributes (covariates) that may affect these concentrations. This led us to the dataset titled "Beijing PM2.5 Data" from the UCI Machine Learning Repository , which has 13 covariates with 43824 observations over 5 years from 2010 to 2014. The response variable is the PM2.5 concentrations, while the 12 predictor variables include: Year, month, day, hour, dew point, temperature, pressure, wind direction, wind speed, hours of snow, and hours of rain.
Here is a table of the data's first 6 and last 6 observations:

	No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<fctr>	<dbl>	<int>	<int>
1	1	2010	1	1	0	NA	-21	-11	1021	NW	1.79	0	0
2	2	2010	1	1	1	NA	-21	-12	1020	NW	4.92	0	0
3	3	2010	1	1	2	NA	-21	-11	1019	NW	6.71	0	0
4	4	2010	1	1	3	NA	-21	-14	1019	NW	9.84	0	0
5	5	2010	1	1	4	NA	-20	-12	1018	NW	12.97	0	0
6	6	2010	1	1	5	NA	-19	-10	1017	NW	16.10	0	0

	No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<fctr>	<dbl>	<int>	<int>
43819	43819	2014	12	31	18	10	-22	-2	1033	NW	226.16	0	0
43820	43820	2014	12	31	19	8	-23	-2	1034	NW	231.97	0	0
43821	43821	2014	12	31	20	10	-22	-3	1034	NW	237.78	0	0
43822	43822	2014	12	31	21	10	-22	-3	1034	NW	242.70	0	0
43823	43823	2014	12	31	22	8	-22	-4	1034	NW	246.72	0	0
43824	43824	2014	12	31	23	12	-21	-3	1034	NW	249.85	0	0

4 Data Preprocessing

4.1 Handling Missing Values

The dataset suffers from missing values. Out of all 43824 observations of PM2.5 level, there are 2067 (or approximately 4.7%) missing PM2.5 values represented as NA in the dataset. After observation, we found that the missing values all appear in clumps of continuous time periods, indicating the data does not go missing on random. In order to encompass the entire dataset into our model, we decided to approximate the NA's by using R's *na.approx* function. Ideally, the linear interpolation performed by the function would work best with random observations of missing PM2.5 levels. As a result, there are some observations that remain as NA even after the approximation.

It turns out that all but the first 24 missing values are replaced by their linear interpolation values. And in fact, the first 24 NA values appear at the very beginning of the dataset, which we can infer occurs because there is no starting value to conduct the linear interpolation. Instead, to replace these 24 values, the value at the closest data extreme is used, which is the 25th observation value.

4.2 Dealing with Time

The only apparent categorical variable in our dataset is *cbwd*, the wind direction. However, the temporal columns of year, month, day and hour can also be seen as categorical variables.

First, we must determine how we should handle the temporal data. One solution would be to convert all four regressors into categorical ones, but the resulting number of columns for the indicator variables would increase drastically (since there's 5 years, 12 months, 30 days, 24 hrs). From the previous homework, we've seen how modifying it as a categorical variable improves the model over the numerical approach. However, for our case, there is a tradeoff. Furthermore, we are only able to choose one categorical variable to create dummy variables with, since not only will high multicollinearity be likely, the interpretations of the binary values would will also be different.

For the sake of choosing the most appropriate categorical variable, we consider PM2.5 level to be weather-related. For weather data, it usually comes with seasonality. For example, we would expect air pollution to be more severe during the winter due to a heavier coal consumption. Therefore, we decided to only change the categorical predictor of months into dummy variables.

There is a way to do the categorization for more than one categorical variables (take *year* and *month* for example), but this involves creating dummy variables for each combination of year and month, which is $5 * 12 = 60$ new columns. We figured that a model with this many regressors would not be appropriate. As a result, we decided to keep the other categorical variables (except cbwd) as their original numerical values.

The figure below shows the head of the dataset now.

	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
	<int>	<int>	<int>	<int>	<dbl>	<int>	<dbl>	<dbl>	<fctr>	<dbl>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2010	1	1	0	129	-21	-11	1021	NW	1.79	0	0	1	0	0	0	0	0	0	0	0	0	0	0
2	2010	1	1	1	129	-21	-12	1020	NW	4.92	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	2010	1	1	2	129	-21	-11	1019	NW	6.71	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	2010	1	1	3	129	-21	-14	1019	NW	9.84	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	2010	1	1	4	129	-20	-12	1018	NW	12.97	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	2010	1	1	5	129	-19	-10	1017	NW	16.10	0	0	1	0	0	0	0	0	0	0	0	0	0	0

5 The Model

Our final selected model is:
$ihs(pm2.5) = \beta_1 \cdot DEWP + \beta_2 \cdot TEMP + \beta_3 \cdot ihs(Iws) + \beta_4 \cdot ihs(Ir) + \beta_5 \cdot Jan + \beta_6 \cdot Feb + \beta_7 \cdot Mar + \beta_8 \cdot$

$Apr + \beta_9 \cdot May + \beta_{10} \cdot Jun + \beta_{11} \cdot Aug + \beta_{12} \cdot Sep + \beta_{13} \cdot Oct + \beta_{14} \cdot Nov + \beta_{15} \cdot Dec$

The betas are given below:

Regressor	**Beta**
DEWP	0.137
TEMP	0.052
ihs(Iws)	-0.179
ihs(Ir)	-0.179
Jan	7.349
Feb	6.888
Mar	5.756
Apr	4.303
May	2.819
Jun	1.573
Aug	0.909
Sep	1.978
Oct	3.625
Nov	5.415
Dec	6.891

Which is in the form of: $y = \beta_1 x_1 + ... + \beta_{15} x_{15}$
The intercept term was removed to allow for all of the months to be accounted for individually. The model above that we have selected ended up being the majority of the full MLR model.

Initially, to come up with the full model, we decided to first plot correlation plots for all regressors that we believed might have a constant variance.

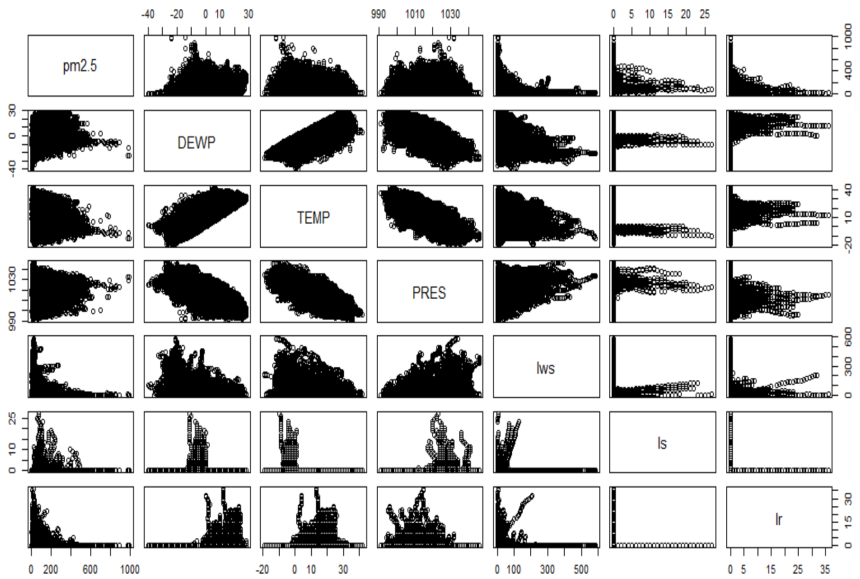

Figure 1: Correlation Plots before transformation

From Figure 1, we notice that the associations between PM2.5 and wind speed(Iws), cumulative hours of snow(Is), and cumulative hours of rain(Ir) all have a rightwards skew and seem to be highly correlated. If we were to leave these covariates as is in the model, then we risk high multicollinearity. One solution we

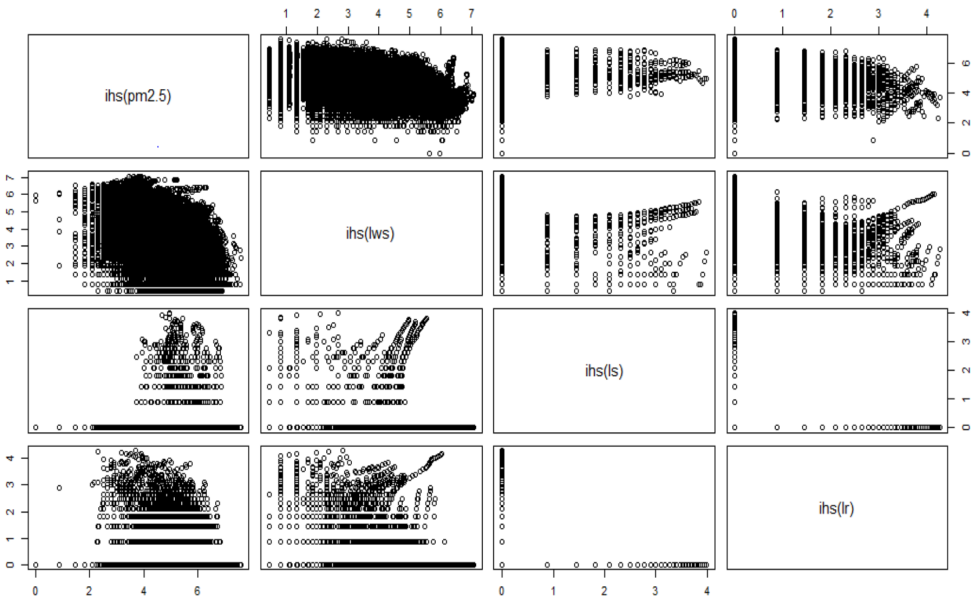

Figure 2: Transformed Correlation Plots

intended to use was a log-transformation on these three covariates, but since wind speed, hours of snow and hours of rain can all be zero and as a result cannot be logged, we chose to use an Inverse Hyperbolic Sine transformation instead. The IHS transformation is defined by

$$f(x) = \log(x + \sqrt{x^2 + 1})$$

The IHS transformation works with data defined on the whole real line including zeros. For large values of x, IHS behaves like a log transformation, and the transformation accommodates values of 0.

After the initial stage of our model selection, we decided to take the IHS of our response, $pm2.5$, which drastically increased the adjusted R^2 and significantly improved the model assumptions. Prior to the variable selection in the section that follows, there was originally another phase of our project that was without the IHS transformation on the PM2.5 response. We decided to only report the model assumptions of those initial findings, which turned out to not be met.

Now that the correlation plots seem to not show any evidence of non-constant variance (Figure 2), we wanted to confirm that having months as the categorical variable to be categorized was indeed the correct choice. Running MLR on the dataset that includes the IHS transformations, we found the categorization of *month* resulted in a better adjusted R^2 than the categorization of the other categorical variables *year* and *cbwd*. The values for these can be found in the next section. This means that our reasoning for the categorizing months was sound.

Afterwards, we decided to add a few interaction terms based on our own knowledge of how atmospheric factors are related, particularly the dew point, pressure, and temperature. We figured they were likely to be highly dependent on each other, and to test whether this dependency would be important in our model, we included the terms: $DEWP*PRES$, $DEWP*TEMP$, $TEMP*PRES$, and $TEMP*PRES*DEWP$.

At this point, the full model includes the response of $ihs(pm2.5)$ and the regressors of $DEWP, TEMP, PRES$, their 4 interactions, $ihs(Iws), ihs(Ir), ihs(Is)$, numerical $day, hour, year$, and 12 binary regressors for each month. So we now can start selecting ideal variables from these to create our most optimal model.

5.1 Variable Selection

Methods that we ended up using to select the best covariates out of the full model included adj. R^2, AIC and BIC through backwards elimination, F-tests of significance from $ANOVA$, VIF to check for multicollinearity, and variable selection by $Lasso$ (to prevent overfitting) with optimal λ from cross-validation.

All of these methods weren't used one after the other in toning down the model, but rather used at different points of our model decisions. Here is a quick overview of our variable selection procedure and thought process:

1. AIC/BIC: removed $year$
2. ANOVA: considered removing $PRES$
3. VIF: removed all interaction terms, $PRES$
4. ANOVA: considered removing $ihs(Is)$
5. AIC/BIC: removed $ihs(Is)$
6. VIF: removed Jul, considered removing $Temp$
7. ANOVA: considered removing Aug
8. LASSO: removed $day, hour$

5.1.1 Adjusted R^2

We obtained several adj. R^2 values to determine which categorical variable ($month$) would be best to use with dummy variables. Adj. R^2 also served as the initial estimate in the difference of fit between two drastically different models, such as the different categorization methods and performing IHS on the response. This was directly found by calling the $summary$ function on our model.

Below are our findings that confirmed the categorization of months was most optimal:

Categorized Variable	adj. R^2 (prior log response)	adj. R^2 (after log response)
Month	0.7229	0.9789
Year	0.6784	0.9747
cbwd	0.6826	0.9765

For the other model changes that we made, if the adj. R^2 only increased/decreased by a small amount then we simply ignored this step, as adj. R^2 naturally increases as more covariates are used, which isn't necessarily a good thing as the model would be prone to overfitting.

The final adj. R^2 value of our model is very high, 0.9679.

5.1.2 AIC and BIC

We also conducted backwards elimination AIC and BIC for each model that we considered, starting with the full model itself.

Running backwards AIC on the full model yielded an AIC of -29040.64 without removing any of the regressors. However, backward BIC yielded an AIC of -28823.44 after removing the $year$ regressor.

The later models that we considered, after refitting the model when $PRES$ was removed, resulted in the removal of $ihs(Is)$, which is the transformed cumulative hours of snow.

5.1.3 Model Analysis: ANOVA and Confidence Intervals

ANOVA was used primarily for its F-tests, in order to tell us which regressors are significant, and are moreso among each other. We were reluctant to remove these regressors at first due to them still being somewhat significant at the 5 percent significance level. However, their p-values were much higher than those of the other regressors. After each observation, we decided to run AIC/BIC and also VIF afterwards to confirm whether or not to actually remove it.

The F-tests of significance from the ANOVA table of our final model is shown below:

Regressor	**F-test p-value**
DEWP	<2e-16
TEMP	<2e-16
ihs(Iws)	<2e-16
ihs(Ir)	<2e-16
Jan	<2e-16
Feb	<2e-16
Mar	<2e-16
Apr	<2e-16
May	<2e-16
Jun	<2e-16
Aug	0.03726
Sep	<2e-16
Oct	<2e-16
Nov	<2e-16
Dec	<2e-16

We observe that from these p-values, the significance of Aug is relatively low compared to the other regressors. However, we must also recognize that it is still significant enough for our model. Our judgment to not remove Aug is confirmed by its low VIF value, as well as AIC/BIC not removing the regressor.

The compact confidence interval shows us that the coefficients for each variable are very accurate. They do not seem very sensitive to the removal of a few regressors, but are extremely sensitive to transformations, especially for the response, $PM2.5$.

	CI lower bound	CI upper bound
	-21.2356663	-21.1152811
DEWP	7.8849015	7.8877668
TEMP	-0.8246666	-0.8212596
Jan	280.1407625	280.2982885
Feb	268.9292043	269.0778351
Mar	218.5709149	218.6988535
Apr	154.5077318	154.6143743
May	88.4792220	88.5685011
Jun	33.4009379	33.4833609
Aug	-6.2326102	-6.1519106
Sep	41.3639899	41.4505185
Oct	131.8313083	131.9315995
Nov	201.7512911	201.8774666
Dec	258.4616195	258.6128293
Iws_n	-8.1431935	-8.1310555
Ir_n	-21.9285562	-21.8869411

Figure 3: Confidence Intervals

5.1.4 Multicollinearity: VIF

Multicollinearity is the presence of a relationship or collinearity between three or more variables which indicates that there could be a redundancy in our data. It might also mean that the data and regression

result is unstable. VIF measures how much a regression coefficient is inflated due to multicollinearity. It is advised to take out dependent variables that have VIFs greater than 10 and to be cautious if it has a VIF greater than 5.
The formula for a VIF score of variable i is:

$$VIF_i = \frac{1}{(1 - R_i^2)}$$

We have calculated the VIF values for our coefficients and found that we had some evidence of multicollinearity in our data, particularly $PRES$, $TEMP$, $PRES$, and the interaction terms. The coefficients for them were extremely high, as shown below.

DEWP	TEMP	PRES	day	hour	ihs(Is)	ihs(Iws)	ihs(Ir)	Jan	Feb
79211.549846	84845.520363	91302.147431	4.235123	4.442585	1.088193	5.189675	1.121613	7807.002001	7095.496422
Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
7774.626832	7496.323628	7733.544775	7487.547959	7754.557855	7750.660130	7510.295410	7799.782503	7552.520936	7803.031103
DEWP:TEMP	DEWP:PRES	TEMP:PRES	DEWP:TEMP:PRES						
61174.606459	78038.773601	83765.606402	59761.579904						

Figure 4: Initial VIF Coefs

The VIF scores caused us to drastically alter our model in order to achieve an acceptable level of multicollinearity. To lower the VIF, we decided to remove $PRES$ and all of the interaction terms. With these variables removed we still had $TEMP$ above the acceptable limit of 10, around a VIF of 12. We then decided to remove the month with the highest VIF to fix this, Jul, and it lowered all VIF scores within the acceptable range. Afterwards we considered removing August to try and have all VIF's below 5. We chose August because ANOVA informed us that it was not as significant as it used to be. However, this did not change the R^2 and the VIF did not lower to the next threshold.

Below is the table of the VIF for the final model:

DEWP	TEMP	ihs(Iws)	ihs(Ir)	Jan	Feb	Mar	Apr
5.183110	7.601582	4.603980	1.087457	1.611690	1.406751	1.452989	1.448260
May	Jun	Aug	Sep	Oct	Nov	Dec	
1.531455	1.584527	1.693381	1.406012	1.225405	1.309465	1.587702	

Figure 5: Final VIF Outputs

5.2 Lasso

Since a larger model is subject to overfitting and high variance, one way of combating this is to introduce regularization through LASSO. LASSO, when compared to ridge regression, allows us to recognize variables that do not contribute to the LASSO regression. In a sense, we can use this as another way to select variables for our MLR model. For our case, we used LASSO at the very end of the model selection process in order to determine whether or not the model originally overfit the data, and if there were some more regressors that we could remove.
First, we took a sample 30-70 partition of the data for cross-validation purposes. This was used to perform

a best-lambda optimization. The best lambda for our LASSO model was 0.02437619, which is quite small. We can interpret this value as such — since the best lambda is an indicator of how much the regularization term affects the fitting of betas, because our lambda is so small, we can say that regularization did not have a substantial effect on the impact of each regressor.

However, after we extracted the lasso model's normalized coefficients using the best lambda, we found that the coefficients for *day* and *hour* were low enough that they had little impact on the actual model itself. Both regressors did indeed improve the model, but by an insignificant margin according to LASSO. Therefore, we decided to remove them from our original model.

Figure 6 shows plots for the lasso lambda optimization which tends to increase MSE and beta coefficients as log(lambda) increases:

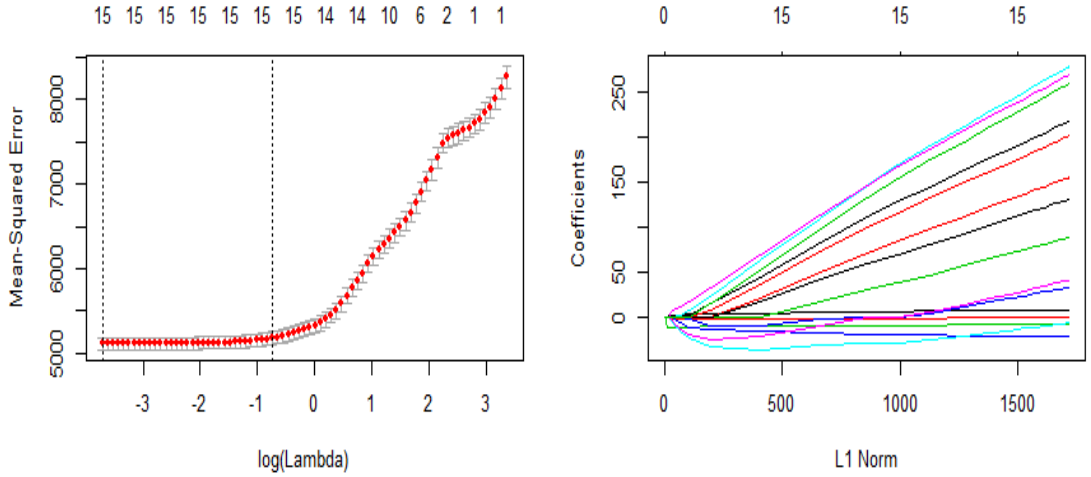

Figure 6: Lasso Lambda Plots

6 Checking Model Assumptions

The assumptions for a linear model are the following: a linear trend with a homogeneity assumption, normality among errors, and low multicollinearity influence to the R^2.
Before the results reported above, we had initially attempted to create our model without performing IHS on the response, $pm2.5$. However, this original model did not meet the normal assumptions at all. In other words, after we worked and got a decent model we realized that our model did not meet the assumptions of normality.
We took measures to make our data normal by transforming our response variable with the IHS transformation. This significantly improved normality. The model still is not perfectly normal, as there is a slight trend on the Residuals vs Fitted Plot, but it is a lot better than what we had before.
As George Box said "Essentially, all models are wrong, but some are useful" in his 1987 book, Empirical Model-Building and Response Surfaces. The QQ-Plot shows that the errors are distributed normally and verifies our normal assumption. Our multicollinearity section above showed how using VIF we reduced the multicollinearity. To summarize, by performing this transformation on the response, the model assumptions improved drastically.
The plots below show the results before and after transforming the response:

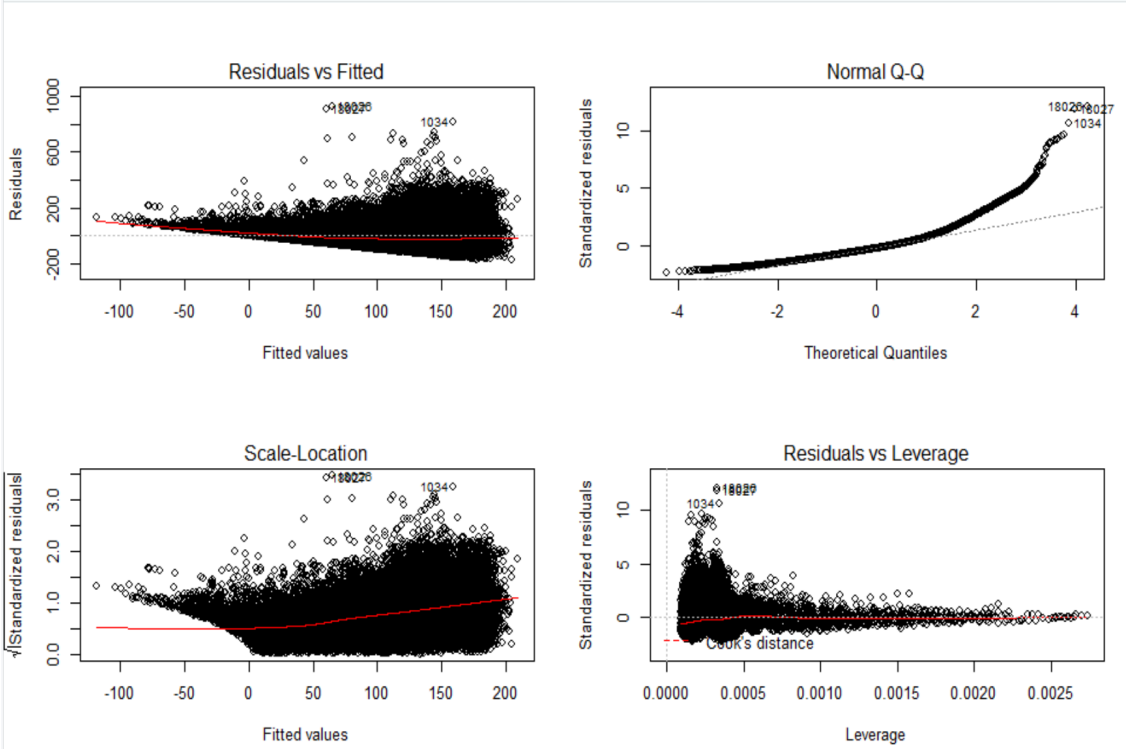

Figure 7: Plots pre-ihs(pm2.5)

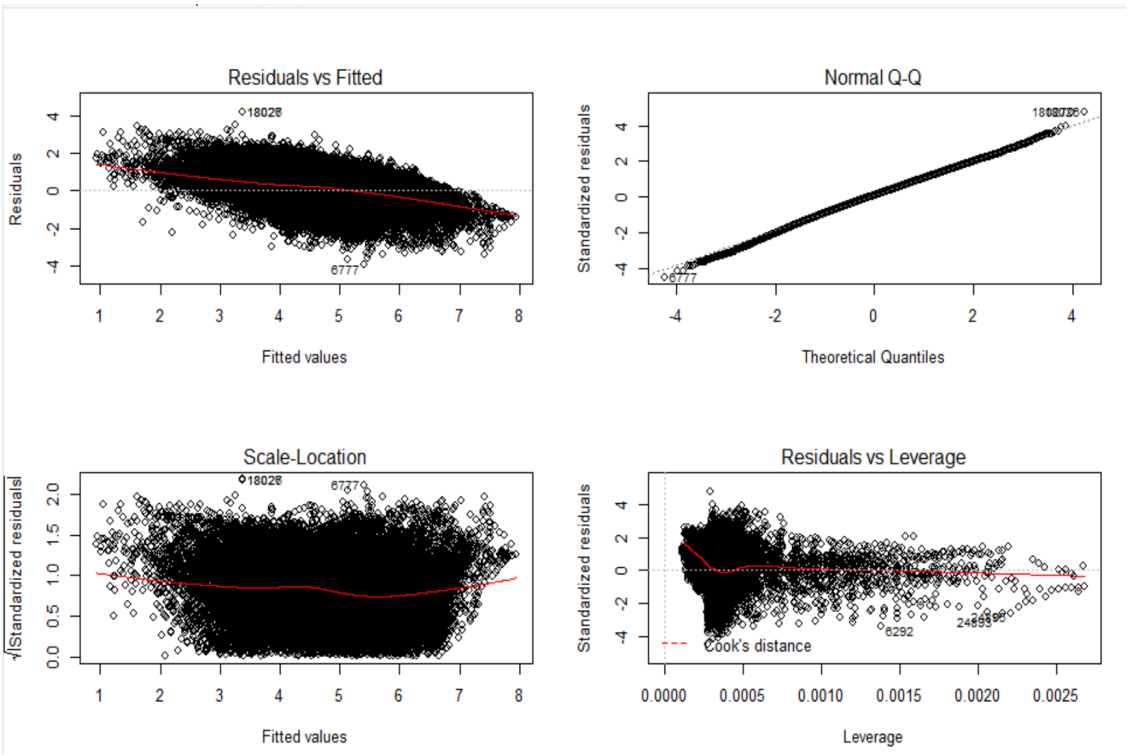

Figure 8: Plots post-ihs(pm2.5)

7 Predictions

As a way to make use of our model, we decided to predict PM2.5 levels for those that were originally NA and we had to approximate through linear interpolation. The below table shows the head of the dataset

(originally NA) that now has the predicted responses:

year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	pm2.5_predicted
<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<fctr>	<dbl>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2010	1	1	0	NA	-21	-11	1021	NW	1.79	0	0	1	0	0	0	0	0	0	0	0	0	0	0	44.16015
2010	1	1	1	NA	-21	-12	1020	NW	4.92	0	0	1	0	0	0	0	0	0	0	0	0	0	0	40.85525
2010	1	1	2	NA	-21	-11	1019	NW	6.71	0	0	1	0	0	0	0	0	0	0	0	0	0	0	42.87612
2010	1	1	3	NA	-21	-14	1019	NW	9.84	0	0	1	0	0	0	0	0	0	0	0	0	0	0	35.96255
2010	1	1	4	NA	-20	-12	1018	NW	12.97	0	0	1	0	0	0	0	0	0	0	0	0	0	0	46.22421
2010	1	1	5	NA	-19	-10	1017	NW	16.10	0	0	1	0	0	0	0	0	0	0	0	0	0	0	59.34351

8 Summary and Discussion

96.79% of the variation in the PM2.5 levels can be explained by our model. To further analyze the choices and results of our model, we can interpret the model selection as well as the betas themselves.

8.1 Model Selection Interpretation:

By applying statistical methods on the PM2.5 data from 2010 to 2014, we have come up with a Multiple Linear Regression model to predict the PM2.5 level in Beijing. Our model eventually used dew point, temperature, wind speed, hours of rain and all months except July as the predictors.

Based on our model, it appears that temperature and pressure are correlated with seasonality to a certain degree. This means we could have taken many approaches to our model selection. Instead of removing $PRES$, we could have easily removed all the months and included $PRES$ instead. However, we thought that by keeping the months, we could provide better analysis and justification for increases in PM2.5 concentrations, as this would involve time.

Removing hours of snowfall makes sense in real life since snowy weather does not occur throughout the year. This means for about 9 months of the year where there is no snow, the hours of snowfall would all be zero, which makes hours of snowfall a bad predictor of PM2.5.

8.2 Beta Interpretation:

Out of every regressor, we only have negative betas for wind speed and hours of rain. This makes sense since if we have strong wind and heavy precipitation, we would expect the PM2.5 concentration to be lower.

For the months, we can see that the largest betas appear in winter seasons, indicating colder weather affects PM2.5 more. This also makes sense in real life. For instance, the need for heating in winter would lead to heavier coal consumption, which would in turn aggravate the air pollution.

8.3 Reflection

In order to battle air pollution, the government of Beijing launched several initiatives to lower the PM2.5 level. One of them is cloud seeding — the weather modification method of increasing the precipitation by artificially adding cloud condensation into the air. In real life, the method is proved to be effective but limited. Rainmaking can only decrease the PM 2.5 level in areas of precipitation temporarily, but the long term effect cannot be assured. The reason people might feel the air is fresher after rain is that precipitation often comes with wind. While wind speed is indeed a predictor of our model and can affect the PM 2.5 level, wind-making is not yet practical for the time being.

In this study, we only looked at the effect meteorological events and seasonality the dataset provided have on the PM2.5 level, while overlooking human activity such as car and factory emission as a determining factor. In future studies, auxiliary datasets can be introduced to account for effects human activities have on the air pollution, such as coal consumption.

9 Appendix

```
library(dplyr)
library(zoo)
beijing <- read.csv("Beijing.csv", header = TRUE)
# for model
head(beijing)
tail(beijing)

# dealing with NAs (approximations)
beijing <- beijing[,-1]
beijing$pm2.5 <- na.approx(beijing$pm2.5, na.rm = FALSE, rule = 2)

# indicator variable columns formation
#beijing <- beijing %>% mutate(Spr = as.numeric(month %in% c(3,4,5)))
#beijing <- beijing %>% mutate(Sum = as.numeric(month %in% c(6,7,8)))
#beijing <- beijing %>% mutate(Fal = as.numeric(month %in% c(9,10,11)))
#beijing <- beijing %>% mutate(Win = as.numeric(month %in% c(12,1,2)))


beijing <- beijing %>% mutate(Jan = as.numeric(month == 1))
beijing <- beijing %>% mutate(Feb = as.numeric(month == 2))
beijing <- beijing %>% mutate(Mar = as.numeric(month == 3))
beijing <- beijing %>% mutate(Apr = as.numeric(month == 4))
beijing <- beijing %>% mutate(May = as.numeric(month == 5))
beijing <- beijing %>% mutate(Jun = as.numeric(month == 6))
beijing <- beijing %>% mutate(Jul = as.numeric(month == 7))
beijing <- beijing %>% mutate(Aug = as.numeric(month == 8))
beijing <- beijing %>% mutate(Sep = as.numeric(month == 9))
beijing <- beijing %>% mutate(Oct = as.numeric(month == 10))
beijing <- beijing %>% mutate(Nov = as.numeric(month == 11))
beijing <- beijing %>% mutate(Dec = as.numeric(month == 12))
beijing <- beijing %>% mutate(Year10 = as.numeric(year == 2010))
beijing <- beijing %>% mutate(Year11 = as.numeric(year == 2011))
beijing <- beijing %>% mutate(Year12 = as.numeric(year == 2012))
beijing <- beijing %>% mutate(Year13 = as.numeric(year == 2013))
beijing <- beijing %>% mutate(Year14 = as.numeric(year == 2014))
beijing <- beijing %>% mutate(SW = as.numeric(cbwd == 'cv'))
beijing <- beijing %>% mutate(NE = as.numeric(cbwd == 'NE'))
beijing <- beijing %>% mutate(NW = as.numeric(cbwd == 'NW'))
beijing <- beijing %>% mutate(SE = as.numeric(cbwd == 'SE'))

# getting rid of categorical columns
#beijing <- subset(beijing, select = -c(cbwd, month))

# again for report
head(beijing)

# ihs transformation
ihs <- function(x) {
  y <- log(x + sqrt(x ^ 2 + 1))
  return(y)
}
```

```
# bad models for report
# categorizing months
ml1 <- lm(pm2.5~ DEWP+ TEMP + PRES + DEWP:TEMP + DEWP:PRES +
            TEMP:PRES + DEWP:TEMP:PRES + day + hour +
            ihs(Is)+ihs(Iws)+ ihs(Ir) + Jan + Feb + Mar + Apr + May + Jun + Jul + Aug+ Sep + Oct+
            Nov + Dec - 1, data = beijing)
summary(ml1)

# categorizing years
ml2 <- lm(ihs(pm2.5)~ DEWP+ TEMP + PRES + DEWP:TEMP + DEWP:PRES +
            TEMP:PRES + DEWP:TEMP:PRES + day + hour + month +
            ihs(Is)+ihs(Iws)+ ihs(Ir) + Year10 + Year11 + Year12 + Year13 + Year14 - 1, data = beijing)
summary(ml2)

# categorizing cbwd
ml3 <- lm(ihs(pm2.5)~ DEWP+ TEMP + PRES + DEWP:TEMP + DEWP:PRES +
            TEMP:PRES + DEWP:TEMP:PRES + day + hour + month + year +
            ihs(Is)+ihs(Iws)+ ihs(Ir) + SW + NE + NW + SE - 1, data = beijing)
summary(ml3)

# trial models
#ml1 <- lm(pm2.5~ DEWP + TEMP + PRES + log(Iws+0.00001) +
        log(Is+0.00001)+ log(Ir+0.00001) + day + hour, data = beijing)
#ml3 <- lm(pm2.5~ DEWP + TEMP + PRES + DEWP:TEMP + DEWP:PRES +
        year + TEMP:PRES + DEWP:TEMP:PRES + ihs(Iws) + ihs(Iws) + ihs(Iws) + day + hour -1, data = beij
#ml4 <- lm(pm2.5~ DEWP + TEMP + PRES + ihs(Iws) + ihs(Iws) +
        ihs(Iws) + day + hour + (Year10-1)+(Year11-1)+(Year12-1)+(Year13-1)+(Year14-1), data = beijing)
#ml2 <- lm(ihs(pm2.5)~ DEWP + TEMP +PRES + ihs(Iws)+ ihs(Is) +
        ihs(Ir) + Spr + Sum + Fal + Win - 1, data = beijing)
#ml1 <- lm(pm2.5~ DEWP  + TEMP + PRES + ihs(Iws)+ ihs(Ir) +
        Feb + Mar + Apr + May +Jul+Aug+ Sep + Nov + Dec - 1, data = beijing)

# optimal model
mlo <- lm(ihs(pm2.5)~ DEWP  + TEMP + ihs(Iws)+ ihs(Ir) +
    Jan + Feb + Mar + Apr + May + Jun +Aug+ Sep + Oct+
    Nov + Dec - 1, data = beijing)
anova(mlo)
summary(mlo)

# AIC BIC- backwards
backAIC <- step(mlo,direction="backward", data=beijing)
backBIC <- step(mlo,direction="backward", data=beijing, k=log(43824))

# VIF
library(car)
vif(mlo)

# Checking Model Assumptions
par(mfrow=c(2,2))
plot(ml2)
abline(v=2*6/45,lty=2)

#  LASSO
x = model.matrix(mlo)
```

```
y = beijing$pm2.5
# setting parameters
library(glmnet)
set.seed(888)
train = sample(1:nrow(beijing), .7 * nrow(beijing))
test = (-train)
ytest = y[test]
lambda <- 10^seq(10, -2, length = 100)
# choosing best lambda
cv.out <- cv.glmnet(x[train, ], y[train], alpha = 1)
plot(cv.out)
bestlam <- cv.out$lambda.min
# lasso predictions
lasso.mod <- glmnet(x[train, ], y[train], alpha = 1, lambda = lambda)
plot(lasso.mod)
lasso.pred <-predict(lasso.mod, s = bestlam, newx = x[test,])
mean((lasso.pred - ytest)^2)
# coefficient analysis
out = glmnet(x, y, alpha = 1, lambda = lambda)
lasso_coef = predict(out, type = "coefficients", s= bestlam)[1:18,]
lasso_coef

# predictions for missing values
beijing2 <- read.csv("Beijing.csv", header = TRUE)
beijing2 <- beijing2[,-1]
beijing2 <- beijing2 %>% mutate(Jan = as.numeric(month == 1))
beijing2 <- beijing2 %>% mutate(Feb = as.numeric(month == 2))
beijing2 <- beijing2 %>% mutate(Mar = as.numeric(month == 3))
beijing2 <- beijing2 %>% mutate(Apr = as.numeric(month == 4))
beijing2 <- beijing2 %>% mutate(May = as.numeric(month == 5))
beijing2 <- beijing2 %>% mutate(Jun = as.numeric(month == 6))
beijing2 <- beijing2 %>% mutate(Jul = as.numeric(month == 7))
beijing2 <- beijing2 %>% mutate(Aug = as.numeric(month == 8))
beijing2 <- beijing2 %>% mutate(Sep = as.numeric(month == 9))
beijing2 <- beijing2 %>% mutate(Oct = as.numeric(month == 10))
beijing2 <- beijing2 %>% mutate(Nov = as.numeric(month == 11))
beijing2 <- beijing2 %>% mutate(Dec = as.numeric(month == 12))
testt <- subset(beijing2, is.na(beijing2$pm2.5))
head(testt)
inv.ihs <- function(x) {
  y <- sqrt((exp(x) - x)^2 - 1)
  return(y)
}
pm2.5_predicted <- inv.ihs(predict(mlo, newdata = testt))

head(cbind(testt, pm2.5_predicted))


#CONFIDENCE INTERVAL START
new_beijing = beijing
new_beijing <- mutate(beijing, Iws_n = ihs(Iws))
new_beijing <- mutate(new_beijing, Is_n = ihs(Is))
new_beijing <- mutate(new_beijing, Ir_n = ihs(Ir))
y = as.matrix(beijing$pm2.5)
```

```
x = as.matrix(new_beijing[,-c(1:3,6:9,16,22:30,32)])
#Design matrix
x = cbind(1,x)
### Calculate the terms x'x and inv(x'x)
xtx = t(x) %*% x
xtxi = solve(xtx)
beta.hat = xtxi %*% t(x) %*% y
hat.matrix = x %*% xtxi %*% t(x)
###  Calculate SSresiduals
ano <- anova(ml2)
SS.res <- ano$`Sum Sq`[ncol(x)]
###  Calculate estimate of sigma^2
n = length(y)
k=15
sigma.squared.hat = SS.res/(n - k + 1)
cov_beta_hat=solve(t(x)%*%x)*sigma.squared.hat
#extract the variance for beta_hat from the covariance matrix
var_beta_hat=diag(cov_beta_hat)
#the 95% CI for beta_hat
alpha=0.05
crit_value=qt(1-alpha/2,df=n-k-1)
CI=cbind(beta.hat-crit_value*sqrt(var_beta_hat), beta.hat+crit_value*sqrt(var_beta_hat))
colnames(CI)=c("CI lower bound", "CI upper bound")
CI
#CONFIDENCE INTERVAL END
```

10 References

George Box in his 1987 book, Empirical Model-Building and Response Surfaces
http://www.science.smith.edu/ jcrouser/SDS293/labs/lab10-r.html This is how we got our response on how to deal with model not fitting after we transformed it.

Beijing Pollution Data from the UCI Machine Learning Repository
https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data This is where we got our actual data from, this is a public portal to many databases.

Information On Pollution and Pm2.5 specifically
https://en.wikipedia.org/wiki/Particulates This gave us background information on the pollution that we worked with and helped us better understand the data.

Information on Cloud Seeding
https://en.wikipedia.org/wiki/Cloud-seeding This gave us information and to work with when talking about what this study could bring and future options for dealing with pollution in Beijing.

IHS Transformation
https://robjhyndman.com/hyndsight/transformations/ This helped give us information about how to handle log transformations with 0's (IHS).