Predicting Atmospheric Fine Aerosol Concentrations using Multiple Regression Analysis

Project Report (Group 20)
STAT141A Fall Quarter 2020

Andrew T. Weakley, Christina De Cesaris, Zheyuan Walter Yu, Seyoung Jung

December 18, 2020

1 Introduction and Objectives

[1] The Interagency Monitoring of PROtected Visual Environment (IMPROVE) network monitors atmospheric fine particulate matter (*a.k.a.*, $PM_{2.5}$) in US national parks to track long-term trends in air quality and visibility [1]. The National Park Service operates and maintains over 150 aerosol samplers at sites across the contiguous US, Alaska, Hawaii, Virgin Islands, and South Korea (**Figure 1**).

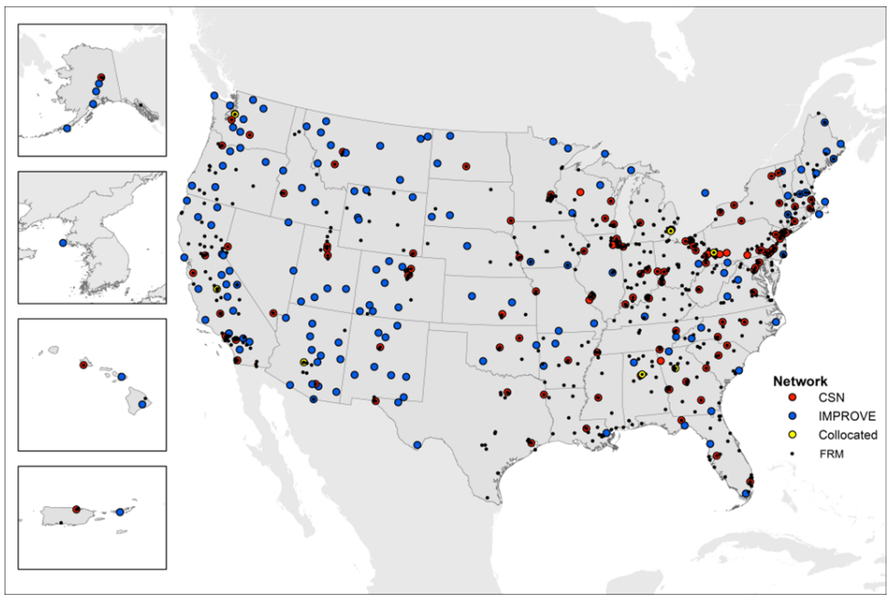


Figure 1: Location of major US particulate monitoring networks. IMPROVE field samplers are denoted as blue dots.

Visibility reduction and the human health impact of $PM_{2.5}$ are related, in a complex way, to both the concentration ($\mu g/m^3$) and composition of atmospheric aerosols [2, 3]. Specifically, variations in regional, seasonal, and local aerosol sources (*e.g.*, summer wildfires in the western US, emission sources related to local industry) obscure a straightforward understanding of the association between total $PM_{2.5}$ collected in the IMPROVE network and the chemical constituents that comprise the aerosol fine mass. Given this, our **first objective** will use publicly available "chemical speciation" records (from the 2015 monitoring year) in a principal component analysis (PCA) to explore the main sources of variability in the speciation data in an effort to attribute variations to emission and aerosol sources.

Our **second objective** will then regresses $PM_{2.5}$ (Y) against the speciation data ($[X]; p <= 32$) using three regression techniques. These methods include multiple linear regression optimized using stepwise variable selection (stepAIC), elastic net regression, and a non-parametric bootstrapped-aggregated (bagged) regression trees [4]. Our **final objective** briefly compares the regression analysis results in the context of the exploratory principal component analysis to judge whether the covariates prioritized for regression analysis may be attributable to emission and atmospheric sources known to compose $PM_{2.5}$ in the IMPROVE network.

[1] Contributions: Section 1: Andrew Weakley; Section 2-4: All team members equally

2 Methods

2.1 IMPROVE 2015 and Github Repository

Data is made available on the Federal Land Management Database managed by Colorado State University (http://views.cira.colostate.edu/fed/). Exactly 21,501 records (N) containing 92 measurements were downloaded. These data correspond to all IMPROVE monitoring sites. Variables may be divided into the following classes as: aerosol carbon (OC, EC), carbon quality-assurance related (OC1, OC2, OC3, OC4, EC1, EC2, EC3, OP), filter light absorbance (fAbs), anions (NO3, SO4), tracer compounds (Al, As, Br, Ca, Cl, Cr, Cu, Fe, Pb, Mg, Mn, Ni, P, K, Rb, Se, Si, Na, Sr, S, Ti, V, Zn, Zr), species constructs (Soil, SeaSalt, ammNitrate, ammSulfate), and sample identifiers (SiteCode, Date). With the exception of fAbs and sample identifiers, all variables are expressed in units of $\mu g/m^3$. Sample analytical uncertainties (*i.e.*, measurement error as standard deviations) were available for all variables.

Metadata were downloaded with pertinent information concerning field sampler location (state and county) and sampler coordinates (longitude, latitude, elevation).

A Github repository is maintained to share and track changes to the database as well as improve communication between team members (https://github.com/cmdecesaris/stats141A-FinalProject).

2.2 Data Cleaning and Sample Partitioning

2.2.1 Variable selection prior to analysis

Data cleaning determined which variables and samples to include in the subsequent analyses. Variables related to carbon quality assurance (with the exception of OP), chemical constructs, and filter light absorbance were all discarded. OP, a quality-assurance variable, was retained for subsequent regression analyses as it is often used as a tracer for specific sources of $PM_{2.5}$ (*e.g.*, wildfire emissions, diesel particulate matter) [5, 6]. Filter absorbance (fAbs) was also removed as it is a spectroscopic proxy for EC and therefore redundant [7].

Species constructs were removed as they consist of mass-weighted linear combinations of the available tracer variables and subject to several (often strict) assumptions as to their atmospheric sources and composition [8]. For example, the "Soil" construct is calculated using the IMPROVE soil equation as $SOIL = 2.2 * AL + 2.49 * SI + 1.63 * CA + 2.42 * FE + 1.94 * TI$ [9]. Here, each coefficient weights Al, Si, Ca, Fe, and Ti based on the most geologically probable amount of inorganic oxygen bonded to each [10]. Essentially, the soil construct is meant to serve as a "representative/typical" US soil sample and my therefore introduce error into our analyses that might otherwise not exist by using individual tracers alone. Overall, 30 chemical measurements (of the carbon, anion, tracer classes) and 2 categorical variables (SiteCode, Date) were available for regression analysis.[2]

2.2.2 Sample selection

Only samples collected in the contiguous US were considered for analysis to ensure that, to the extent possible, $PM_{2.5}$ was as homogeneous as possible, i.e., identically distributed on a site-to-site basis. Unique site identifiers (SiteCode) from South Korea, Alaska, Hawaii, and the Virgin Islands were therefore identified using IMPROVE metadata. The dplyr filter function then screened sites at those states (and country) from the main array leaving $N = 20629$ records for analysis.[3]

Samples exhibiting inordinately negative $PM_{2.5}$ concentrations were also removed using sample analytical uncertainty (σ_i). As $PM_{2.5}$ can never, in principle, be negative, samples with $PM_{2.5} < -3\sigma_i$ were judged has having a high probability of not being zero (a blank measurement). In other words, negative concentrations outside 3-times analytical uncertainty were more likely the result of systematic as opposed to the random measurement error in $PM_{2.5}$. We should further note that this screening criterion is very conservative as this threshold is approximately 2 times less than reported $PM_{2.5}$ minimum detection limits [11]. After screening other records for missing values, the total number of samples available for analysis was $N = 17294$

2.2.3 Stratified Sample Partitioning

Prior to regression analysis, the data were partitioned into training and testing sets by arranging samples alphabetically by SiteCode and Date variables (MM/DD/2015). Once sorted, the data were

[2]The Soil and Sea Salt constructs did aid exploratory analysis given the pattern of observed loadings.

[3]To address concerns in our project proposal, we decided not to use the percentage of samples below method detection limits as a criterion to select variables for analysis given concerns about censoring. More rational criteria are presented below.

split in half by placing every other sample in the test set, using seq(1,n,2), with $N_{train} = N_{test} = 8647$. Random sampling was eschewed in favor of this approach as pollution data shows seasonality, making it critical that seasonal variability in chemical composition are represented equally in the training and testing sets.

2.3 Exploratory Correlation and Principal Component Analysis

Following sample correlation analysis and visualization using ggcorrplot [12], the main sources of variation within the (non-categorical part of the) predictor matrix were explored with principal component analysis (PCA). Variables were centered, but not scaled, prior to analysis as the subsequent regression analyses were performed on non-standardized covariates. The number of major principle components was selected using a scree plot, aided by the use of statistical aprroaches available in the PCATools package [13]. The first several eigenvectors were plotted to determine if the pattern of loadings were attributable to known aerosol sources. Score plots for the major components were then generated and pseudo-colored on relevant species concentrations to aid interpretation.

2.4 Multiple Regression and stepAIC optimization

A first-order multiple regression used the lm() function to regress half the $PM_{2.5}$ (Y) measurements against the 30 speciation measurements ([X]; N_{train}=8647, p=30). Assumptions of linearity, normality and homoscedasticity were checked based on the residual plots, normal Q-Q plots, and Box-Cox procedure. Forward selection, backward elimination, forward stepwise, and backward stepwise selection were toggled in the stepAIC function and each explored as a means of model optimization. The AIC and BIC were then used to selected the "best" eight candidate models according to these criteria. Model validation then used the test data (N_{test}=8647) to select a final model according to overall prediction ability (e.g., RMSE,R^2), consistency between training and testing set performance, and the principle of parsimony (Occam's razor).

2.5 Elastic Net Regression Analysis

Elastic net regression proceeded in a manner similar to lm() fitting, using the cv.glmnet() function available in the glmnet package [14]. Elastic net is appealing for our purposes given that biased estimators typically show better out-of-sample prediction performance than unbiased linear estimators [4]. Specifically, elastic net regression minimizes the least squares objective function (SSE) subject to a LASSO and ridge penalty as

$$\hat{\beta} = \underset{\beta}{argmin}\Big\{ \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p}(\alpha\beta_j^2 + (1-\alpha)|\beta_j|\Big\} \tag{1}$$

where β_j are the p regression coefficients, x_{ij} are the p predictors, λ is the regularization penalty, and α is the elastic net penalty [4]. Elastic net regression balances fidelity to the full-predictor ridge solution ($\alpha = 1$) and the sparse-variate LASSO regression ($\alpha = 0$), the later performing subset selection during estimation.

In practice, the user must specify λ and α. In this study, ten-fold cross validation was used to find the optimal values for λ for 20 different values of α. The mean squared error of prediction (MSEP) was used to evaluate the best model.

2.6 Bagged Regression Trees

Decision tree regression analysis was carried out using the rpart and caret package [15, 16]. Decision trees are highly interpretive flow chart models useful for predicting data based on a series of binary splits constructed through the training data. As a result, decision trees are subject to high variability between models given the same data set. For a single tree model, the training data was partitioned using 10-fold cross validation process to optimize the model. The optimal hyper-parameters were evaluated through grid search and determined based on which resulted in the lowest validation error and Mallow's C_p.

To combat the high variability produced by a single tree model, a tree bagging ensemble method was employed utilizing the caret package. In this process, random subsets of the training data are selected with replacement and used to build a ensemble of different tree models. The predictive ability of the bagged regression tree model is the average perform ace of all trees within that model.

3 Results and Discussion

3.1 Exploratory Analysis

Figure 2 (LHS) shows a plot of the species sample correlations where their axis labels have been organized using the agglomerative hierarchical clustering option in the ggcorrplot function. Notably, the position and clustering of the variables on the correlation plot makes sense given the sources of aerosol most likely impacting IMPROVE samples. Beginning from the bottom left and working to the top right, we see that MG, CL, and NA cluster together and are highly correlated. These species are all good tracers for marine aerosols (sea spray) [17, 18]. Next, we see that carbon species (OC,EC,OP) cluster with $PM_{2.5}$. This likely indicates that carbon comprises as a large amount of $PM_{2.5}$ mass. Indeed, we see that OC, and to a lesser extent EC and OP, comprise a large amount of sample mass in in these samples (on average; see **Figure 2 (RHS)**). Continuing further, we see that species attributable with soil (Al,Si,Fe,Ti), shipping (Ni,V), coal combustion (Se,S,SO4), and incinerator sources (Pb,Zn) all cluster together [9, 19, 17].

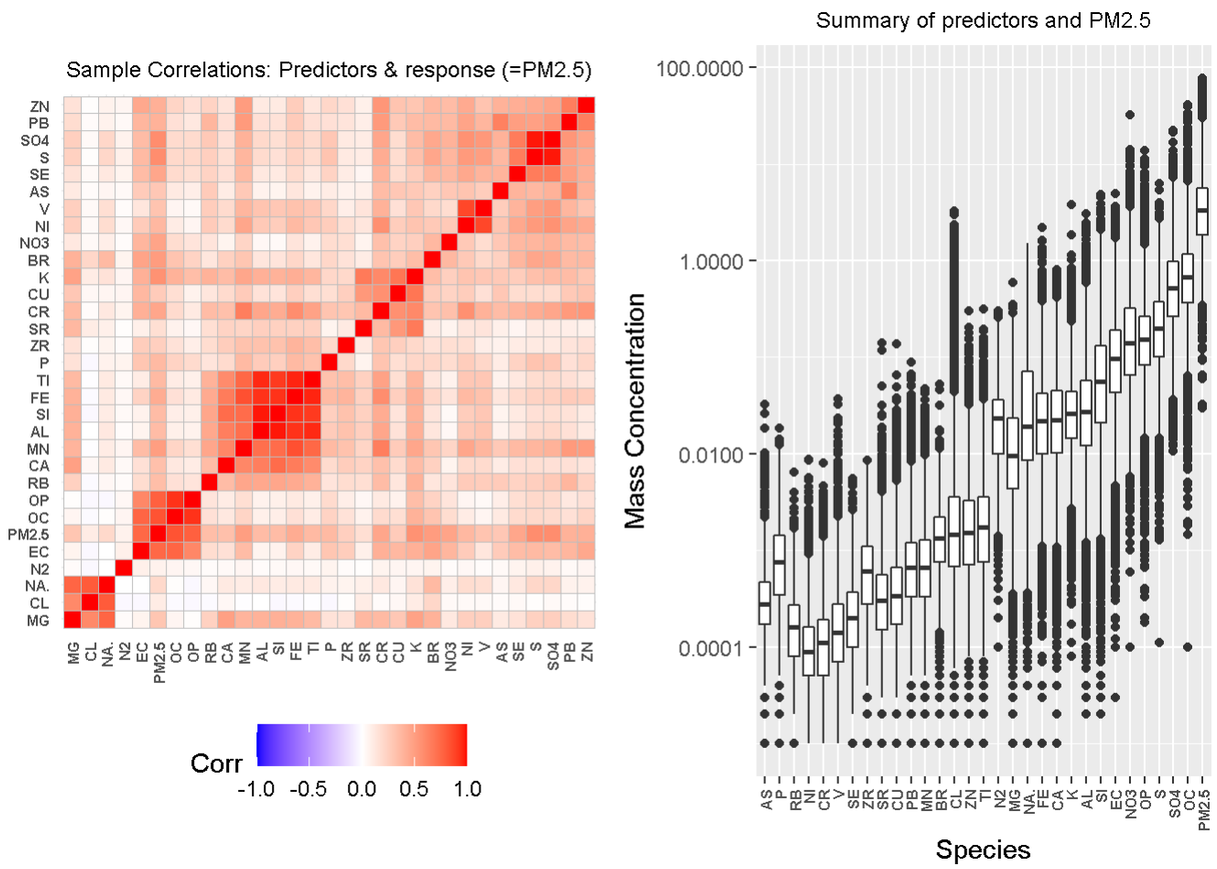


Figure 2: Correlation Matrix and Boxplots of Species comprising $PM_{2.5}$, sorted in order of least mass contribution

Figure 3 shows a scree plot of the individual and cumulative amount of variance explained the in predictor matrix. Here, we see that the first 3 components explain over 95% of the variance in the data. Depending on the component selection method used, 3 to 4 components were considered appropriate to model the data.

After interpreting the principal component loadings (see supplemental material), it was determined that the first and second principal components were likely modeling variance related to organic carbon (OC) and total anion concentration (SO4+NO3) respectively. **Figure 4** shows the sample observations projected onto the first two components, colored according to OC and total anion content, respectively.

Repeating the procedure indicated that the 3rd and 4th components were modeling a rough contrast between NO3 and SO4 (PC3) and soil impacts (see Section S1 in Supplemental Material). Additionally, PC5 and PC6 (more roughly) modeled OP and marine aerosol impacts (see Supplemental Material). Notably, the clear interpretation of the components was directly proportional to the amount of variance explained by each.

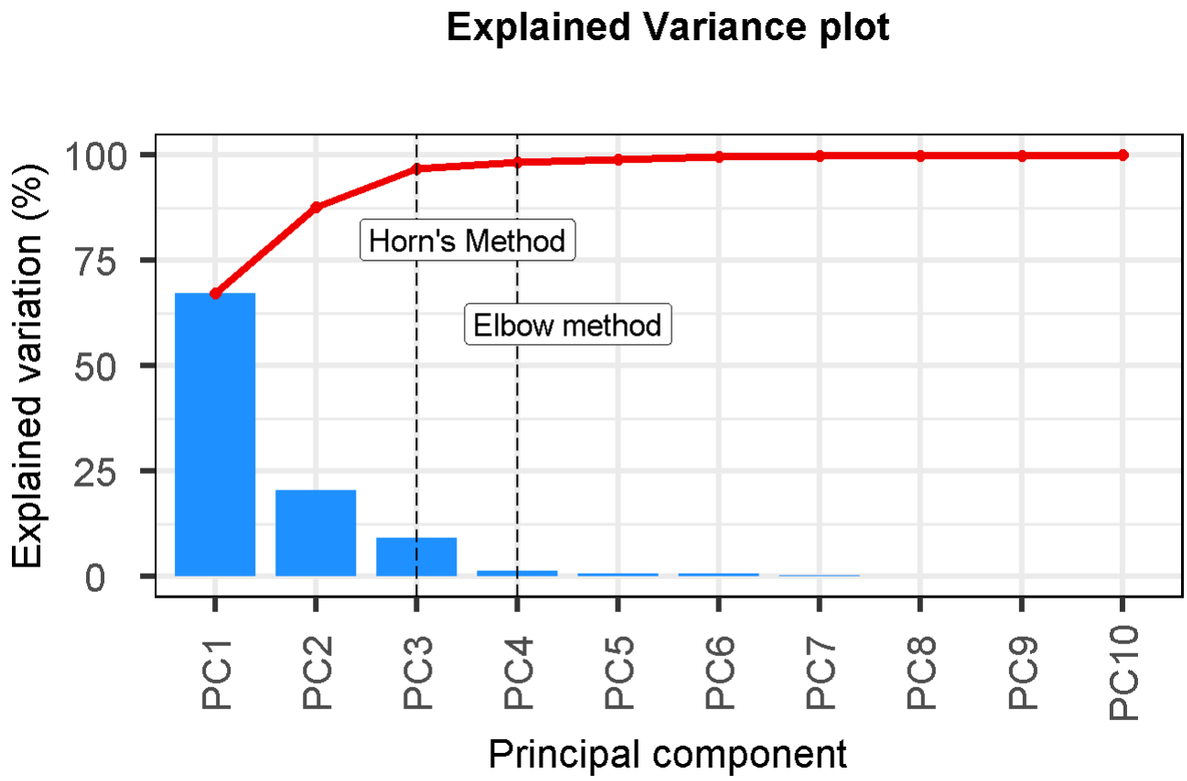


Figure 3: Principal component scree plot

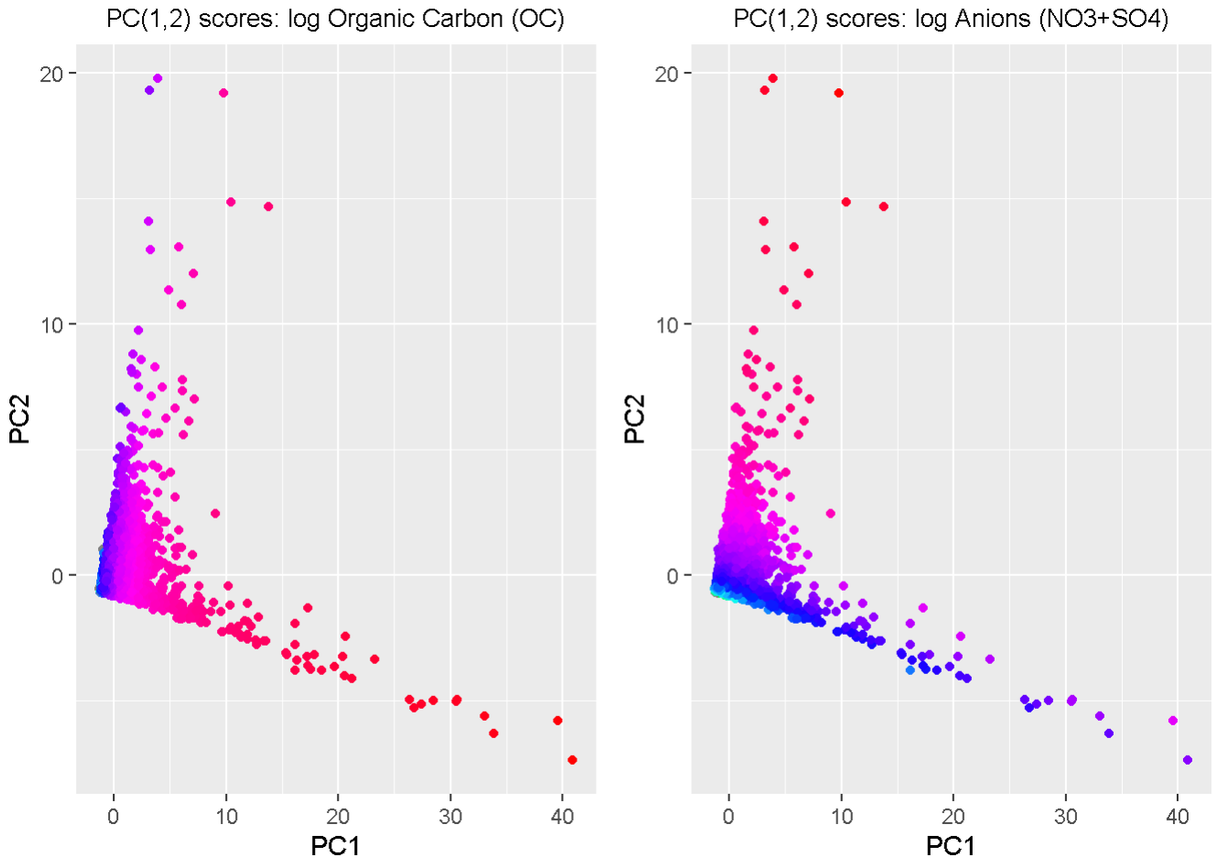


Figure 4: Principal component 1-2 subspace colored on log species concentrations. Blue denotes low species' concentration while magenta denotes higher concentration. log scaling was used to promote contrast.

3.2 Regression analysis

3.2.1 Multiple Regression and stepAIC optimization

Based on residual plots, normal q-q plots, and a Box Cox procedure applied to the eight best stepAIC models, the assumptions of linearity, normality, and homoscedasticity are all valid (see Supplemental Material for details). First, it was interesting that all of these models had very good prediction ability. The R-square values, representing the "variance explained" between the predicted and observed test set values, were very high in these models ($R^2 > 0.977$). The root mean squared error, a measure of the average prediction error made by the model in predicting the outcome for an observation, were very low in all models ($RMSEP < 0.773/m^3$).

The consistency between the models from training data and testing data were also compared. The regression coefficients were very consistent between training data and testing data, because they had the same signs and the similar values.

At the end, the Occam's razor principle was ultimately used to determine the final models. Model 5 and model 7 had the least number of regression coefficients, so they were selected; in fact, these two models were actually the same upon further inspection. Notably, model 5 was selected by forward selection based on BIC, and model 7 selected by forward stepwise based on BIC.[4] The model is given below as:
PM2.5 = - 0.2068 + 1.9302*OC + 0.4269*SO4 + 2.2615*FE + 1.2239*NO3 +
3.5397*CL + 2.9903*SI + 3.8678*S + 2.4363*K + 1.9114*CA - 29.5285*CU +
24.5011*PB + 47.1501*P + 0.2289*OP + 18.4563*TI + 141.3513*SE

3.2.2 Elastic Net Regression

After converting the factor variables (SiteCode and Date) to dummy variables, the training set will have 308 variables (30 + 278 factor levels). Ten-fold cross validation for 20 different values for α (between 0 and 1) was fitted using cv.glmnet. The the test set mean squared error for each fit is given (**Table 1**).

Table 1: Test set MSE for each *alpha* considered by 10-fold CV

Alpha	MSE	Alpha	MSE
0.00	0.7634257	0.55	0.6141631
0.05	0.6054706	0.60	0.6186419
0.10	0.6065652	0.65	0.6227428
0.15	0.6123570	0.70	0.6171354
0.20	0.6071487	0.75	0.6373969
0.25	0.6166402	0.80	0.6155697
0.30	0.6084149	0.85	0.6355489
0.35	0.6113124	0.90	0.6405099
0.40	0.6070638	0.95	0.6545169
0.45	0.5977094	1.00	0.6332386
0.50	0.6352535	-	-

In Table 1, we can see that neither ridge Regression nor LASSO Regression ($\alpha = 0$) gives us the best result. Although both models given very similar results in terms of MSE, the lowest MSE is achieve from α=0.45. Since we are using elastic net Regression, we can expect that this model has fewer predictor variables than the full model (but likely more than the suboptimal LASSO solution). For the model with α=0.45, cross validation chose a λ=0.074.

Indeed, **Table 2** indicates that 52 out of the original 308 variables are nonzero indicating that most of the variables are not useful to the regression problem.

[4]Comprehensive analysis details in Section S2 of supplemental material

Table 2: Elastic net regression coefficients for $(\lambda, \alpha) = (0.074, 0.45)$

Variable	Coefficient	Variable	Coefficient	Variable	Coefficient
(Intercept)	-0.114674053	TI	9.062758625	Date6/11/2015	0.284272033
EC	0.426403466	V	18.565709821	Date6/23/2015	0.044890657
OC	1.711304246	NO3	1.178011209	Date6/29/2015	0.054805817
OP	0.807417478	SO4	0.618401342	Date7/2/2015	0.318254665
AL	0.971613825	SiteCodeCORI1	-0.010450120	Date7/29/2015	0.186479582
BR	12.595389511	SiteCodeELDO1	-0.136389716	Date7/5/2015	0.026638887
CA	1.756914687	SiteCodeMAKA2	0.002197642	Date8/10/2015	0.003149496
CL	2.745861899	SiteCodeMAVI1	0.088369079	Date8/13/2015	0.003635989
FE	3.510792738	SiteCodePHOE1	-0.127508792	Date8/16/2015	0.157029484
PB	8.392527873	SiteCodePORE1	0.660387523	Date8/19/2015	0.288633654
MG	1.062569693	SiteCodeREDW1	0.017723498	Date8/25/2015	0.227724937
P	27.117590226	SiteCodeSAGA1	-0.103433378	Date8/28/2015	0.222549734
K	1.768529297	SiteCodeSAMA1	0.074553728	Date8/31/2015	0.123963928
RB	85.508506989	Date3/13/2015	-0.011986774	Date8/4/2015	0.195171963
SE	127.200123974	Date3/19/2015	-0.026328360	Date8/7/2015	0.147065416
SI	2.285695803	Date3/22/2015	-0.076507845	Date9/18/2015	0.027475458
NA	0.362930418	Date3/4/2015	-0.009827235	Date9/3/2015	0.091789732
S	3.102419694	Date3/7/2015	-0.258708631	-	-

Notably, many of the chemical and tracer variables (21 out of 30) remain in the model.[5] However, a majority of the SiteCode factor levels are dropped and half of their coefficient values are negative. Furthermore, we can see that the selected Date levels are mostly from the summer season (from June to August). This is likely due to the higher PM concentrations in the summer months (At some sites) due to wildfires [20]. At first glance, it interesting to see that a few of the selected variables, such as SE (Selenium), RB (Rubidium), P (Phosphorous) and BR (Bromine), have extremely larger coefficient values compared to the others. However, as shown in **Figure 1** these variables comprise a very small amount of the total mass of $PM_{2.5}$, approximately two orders of magnitude less than OC and SO4 for example. They are therefore given largest coefficients to account for this difference in scale.

3.2.3 Bagged Regression Trees

After performing grid search for tree fitting, the optimal single (non-bagged) tree model had a c_p of 0.1, a minimum split of 10, and a maximum depth of 8. The optimal tree had a RMSE of 2.156 $\mu g/m^3$, an R Squared Value of 0.83, and an MAE of 1.21 $\mu g/m^3$. A graphical depiction of the model is presented in **Figure 6**.

The graphical depiction of the bagged tree model is complex and difficult to show. The final model consisted of 25 tree models, had an RMSE of 1.80 $/m^3$, an R Squared Value of 0.875, and MAE of 0.958 $/m^3$. The bagged tree model was used to plot predictors by importance, normalized to the most important. The most important predictor was carbon, followed by OP, and Potassium. The full break down is presented in **Figure 7**.[6]

[5] Comprehensive analysis details in Section S3 of supplemental material
[6] Comprehensive analysis details in Section S4 of supplemental material

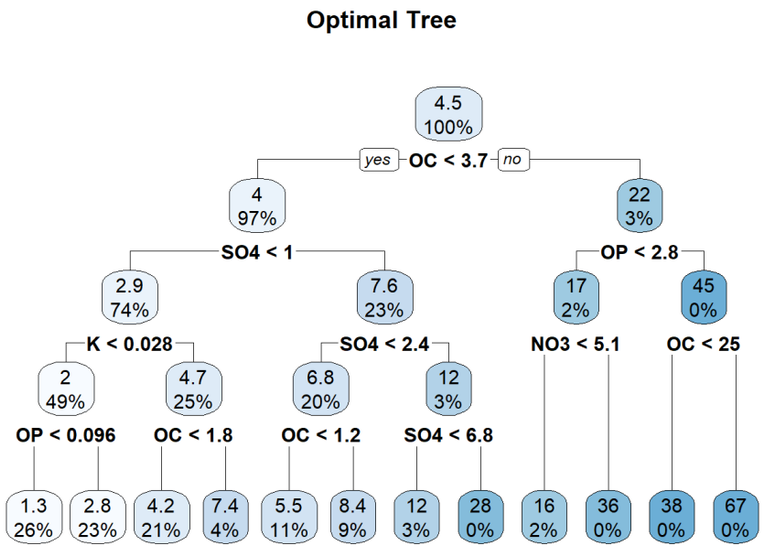


Figure 5: Optimal Decision Tree

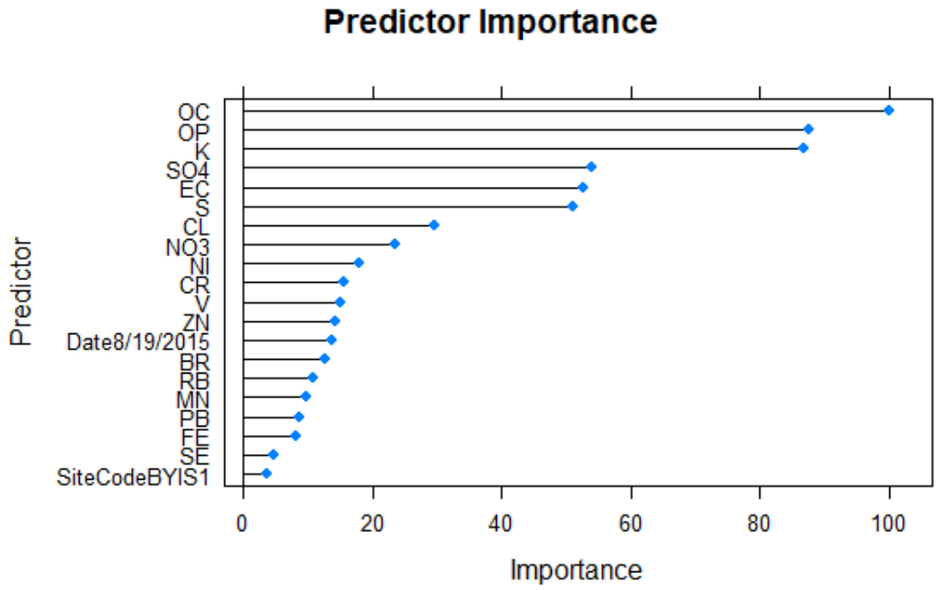


Figure 6: Optimal Decision Tree

4 Conclusions: Model comparison and source attribution

The prediction of $PM_{2.5}$ varied by model and method with test set R^2 ranging from 0.875 in the case of bagged regression trees to as high as 0.977 for multiple regression. Irrespective of the relative prediction performance of each model, we see that they all tended to prioritize very similar covariates. Specifically, carbon predictors (OC,EC,OP) were used in all regressions, with the importance of OC occupying the top spot in both the multiple regression and regression tree following optimization. Furthermore, SO4, OP, and NO3 also occupied principal positions in linear regression and bagged tree. Such results make sense in the context of Figure 1: the principal source of variance in the predictor matrix are from OC and anion content (NO3+SO4).[7]

One notable difference between the parametric regressions (*i.e.*, multiple regression and elastic net) and the regression tree is the former's reliance on carbon and soil-related species (Ca,Si,Ti) while the latter used no tracers for soil species. This reason my have contributed to the regression tree's less-optimal performance when judged against the other models. Overall, all models appear reasonable for the prediction of PM2.5 in the IMPROVE network.

[7]It should be noted that $OC = OC1 + OC2 + OC3 + OC4 + OP$ and $EC = EC1 + EC2 + EC3 - OP$. OP was included in the model as an empirical factor to adjust for potential biases related to OC/EC thermal analysis [21].

References

[1] Paul A. Solomon et al. "U.S. National PM2.5 Chemical Speciation Monitoring Networks—CSN and IMPROVE: Description of networks". In: *Journal of the Air Waste Management Association* 64.12 (2014), pp. 1410–1438. ISSN: 1096-2247. DOI: 10.1080/10962247.2014.956904.

[2] John G Watson. "Visibility: Science and regulation". In: *Journal of the Air Waste Management Association* 52.6 (2002), pp. 628–713. ISSN: 1096-2247.

[3] Cliff I. Davidson, Robert F. Phalen, and Paul A. Solomon. "Airborne Particulate Matter and Human Health: A Review". In: *Aerosol Science and Technology* 39.8 (2005), pp. 737–749. ISSN: 0278-6826. DOI: 10.1080/02786820500191348.

[4] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, 2013.

[5] B. Khan et al. "Differences in the OC/EC ratios that characterize ambient and source aerosols due to thermal-optical analysis". In: *Aerosol Science and Technology* 46.2 (2012), pp. 127–137. DOI: 10.1080/02786826.2011.609194.

[6] Andrew T. Weakley et al. "Ambient aerosol composition by infrared spectroscopy and partial least squares in the chemical speciation network: Multilevel modeling for elemental carbon". In: *Aerosol Science and Technology* 52.6 (2018), pp. 642–654.

[7] Warren H. White et al. "A critical review of filter transmittance measurements for aerosol light absorption, and de novo calibration for a decade of monitoring on PTFE membranes". In: *Aerosol Science and Technology* 50.9 (2016), pp. 984–1002. ISSN: 0278-6826. DOI: 10.1080/02786826.2016.1211615.

[8] JudithC Chow et al. "Mass reconstruction methods for PM2.5: a review". In: *Air Quality, Atmosphere Health* 8.3 (2015), pp. 243–263. ISSN: 1873-9318. DOI: 10.1007/s11869-015-0338-3.

[9] J. L. Hand et al. "Trends in remote PM2.5 residual mass across the United States: Implications for aerosol mass reconstruction in the IMPROVE network". In: *Atmospheric Environment* 203 (2019), pp. 141–152. ISSN: 1352-2310. DOI: https://doi.org/10.1016/j.atmosenv.2019.01.049.

[10] William C. Malm et al. "Spatial and seasonal trends in particle concentration and optical extinction in the United States". In: *Journal of Geophysical Research: Atmospheres* 99.D1 (1994), pp. 1347–1370. DOI: doi:10.1029/93JD02916.

[11] Analytical Methods Committee. "Recommendations for the definition, estimation and use of the detection limit". In: *Analyst* 112.2 (1987), pp. 199–204.

[12] Alboukadel Kassambara. *ggcorrplot: Visualization of a Correlation Matrix using 'ggplot2'*. R package version 0.1.3. 2019. URL: https://CRAN.R-project.org/package=ggcorrplot.

[13] Kevin Blighe and Aaron Lun. *PCAtools: PCAtools: Everything Principal Components Analysis*. R package version 2.2.0. 2020. URL: https://github.com/kevinblighe/PCAtools.

[14] Noah Simon et al. "Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent". In: *Journal of Statistical Software* 39.5 (2011), pp. 1–13.

[15] Terry Therneau and Beth Atkinson. *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-15. 2019. URL: https://CRAN.R-project.org/package=rpart.

[16] Max Kuhn. "Building Predictive Models in R Using the caret Package". In: *Journal of Statistical Software, Articles* 28.5 (2008), pp. 1–26. ISSN: 1548-7660. DOI: 10.18637/jss.v028.i05. URL: https://www.jstatsoft.org/v028/i05.

[17] Xin-Hua Song, Alexandr V. Polissar, and Philip K. Hopke. "Sources of fine particle composition in the northeastern US". In: *Atmospheric Environment* 35.31 (2001), pp. 5277–5286.

[18] Li-Jun Zhao et al. "Magnesium Sulfate Aerosols Studied by FTIR Spectroscopy: Hygroscopic Properties, Supersaturated Structures, and Implications for Seawater Aerosols". In: *The Journal of Physical Chemistry A* 110.3 (2006), pp. 951–958. ISSN: 1089-5639. DOI: 10.1021/jp055291i.

[19] Mar Viana et al. "Impact of maritime transport emissions on coastal air quality in Europe". In: *Atmospheric Environment* 90 (2014), pp. 96–105.

[20] D. M. Murphy et al. "Decreases in elemental carbon and fine particle mass in the United States". In: *Atmospheric Chemistry and Physics* 11.10 (2011), pp. 4679–4686.

[21] Judith C Chow et al. "The DRI thermal/optical reflectance carbon analysis system: description, evaluation and applications in US air quality studies". In: *Atmospheric Environment. Part A. General Topics* 27.8 (1993), pp. 1185–1201. ISSN: 0960-1686.

STA141A-ATW-Markdown

Andrew T. Weakley

12/15/2020

— Step 1: Data loading and procressing —

```
## --- Part a: Upload Metadata for samples ---
path_data<-file.path(getwd(),"data")
META_DATA<-as_tibble(read.csv(file.path(path_data,"IMPROVE_metadata.csv")))
## --- Filter samples from Korea and Canada ---
US_META<-META_DATA %>% filter(Country %nin% c("KR","CA"))


## --- Filter stats not in continental US ---
US_META<-META_DATA %>% filter(State %nin% c("HI","AK","VI"))

## -- Use Mississippi River as a dividing point for WEst-East US --
MR_coords<-c(47.239722, -95.2075)
POS_Sampler<-as.numeric(US_META$Longitude <MR_coords[2])
# --- 1 are WEst US, 0 are East
US_META<-add_column(US_META,WE_US = POS_Sampler)

## --- Part b: Load samples data ---
DATA<-as_tibble(read.csv(file.path(path_data,"IMPROVE_2015_data_w_UNC_v2.csv")))

## --- Part c: Select samples from SW given site identifiers from SW_META table ("C
ode")
US_DATA_all<-as_tibble(DATA %>% filter(SiteCode %in% US_META$Code))
```

```
# Let's identify any samples that (grossly) violate PM2.5 mass balances
# PM2.5 (=Y) cannot be negative!
# Since there's some probability that PM2.5 is negative due to errors at low concen
tration, we may use PM2.5 uncertainties to remove samples that fall outside -3*PM2.
5_UNC.
# In this way, we don't risk censoring the data but do remove likely erroneous dat
a.
US_DATA_all<-US_DATA_all %>% dplyr::filter(PM2.5 > -3*PM2.5_UNC)
```

```
exclude<-c("PM10","POC","ammNO3","ammSO4","SOIL","SeaSalt","OC1","OC2","OC3","OC
4","EC1","EC2","EC3","fAbs_MDL","fAbs")
US_DATA_LRG<- US_DATA_all %>% dplyr::select(!contains(exclude) & !matches("_UNC") |
matches("PM2.5_UNC"))
any(is.na(US_DATA_LRG))
```

```
## [1] TRUE
```

```
US_DATA_LRG<-US_DATA_LRG[which(complete.cases(US_DATA_LRG)),]
any(is.na(US_DATA_LRG))
```

```
## [1] FALSE
```

```
## --- Instead of random partitioning, I will partition by first sorting samples by
SiteCode and DATE (already done) and place every other sample in the test set.
# --- This data has seasonality. Sorting by date therefore ensures seasonality is e
quivalent between datasets
n<-nrow(US_DATA_LRG)
ind_test<-seq(1,n,2)
US_DATA_LRG_test<-US_DATA_LRG[ind_test,]
US_DATA_LRG<-US_DATA_LRG[-ind_test,]
```

— Step 2: Descriptive prior to GMM —

```
# --- Plot of abs and EC ---
ggplot(US_DATA_LRG,aes(x=SiteCode,y=PM2.5,color=SiteCode))+
  geom_boxplot()+
  theme(plot.title=element_text(hjust = 0.5))+
  scale_y_log10(limits=c(0.001,100))+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1,size=4))
```

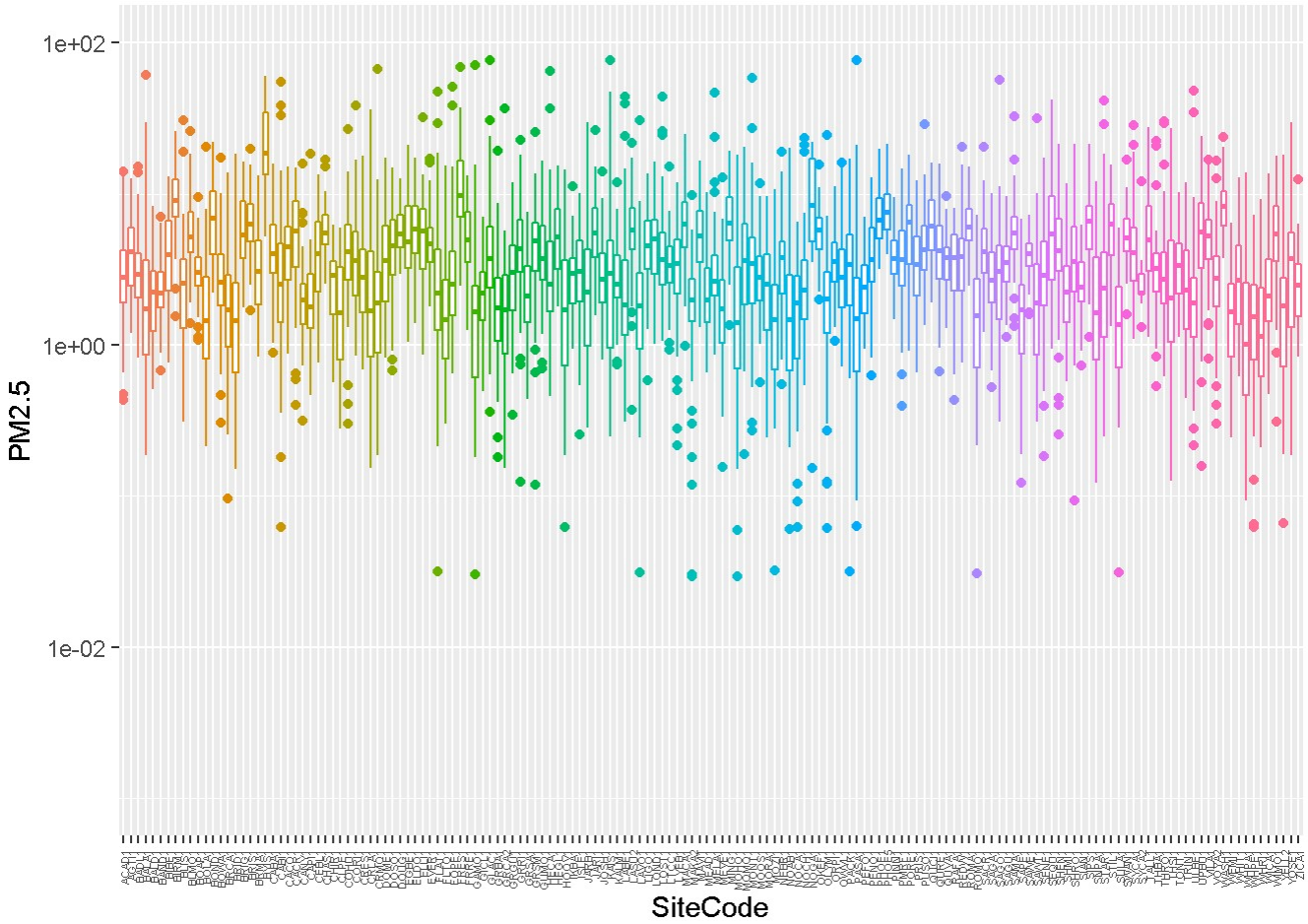
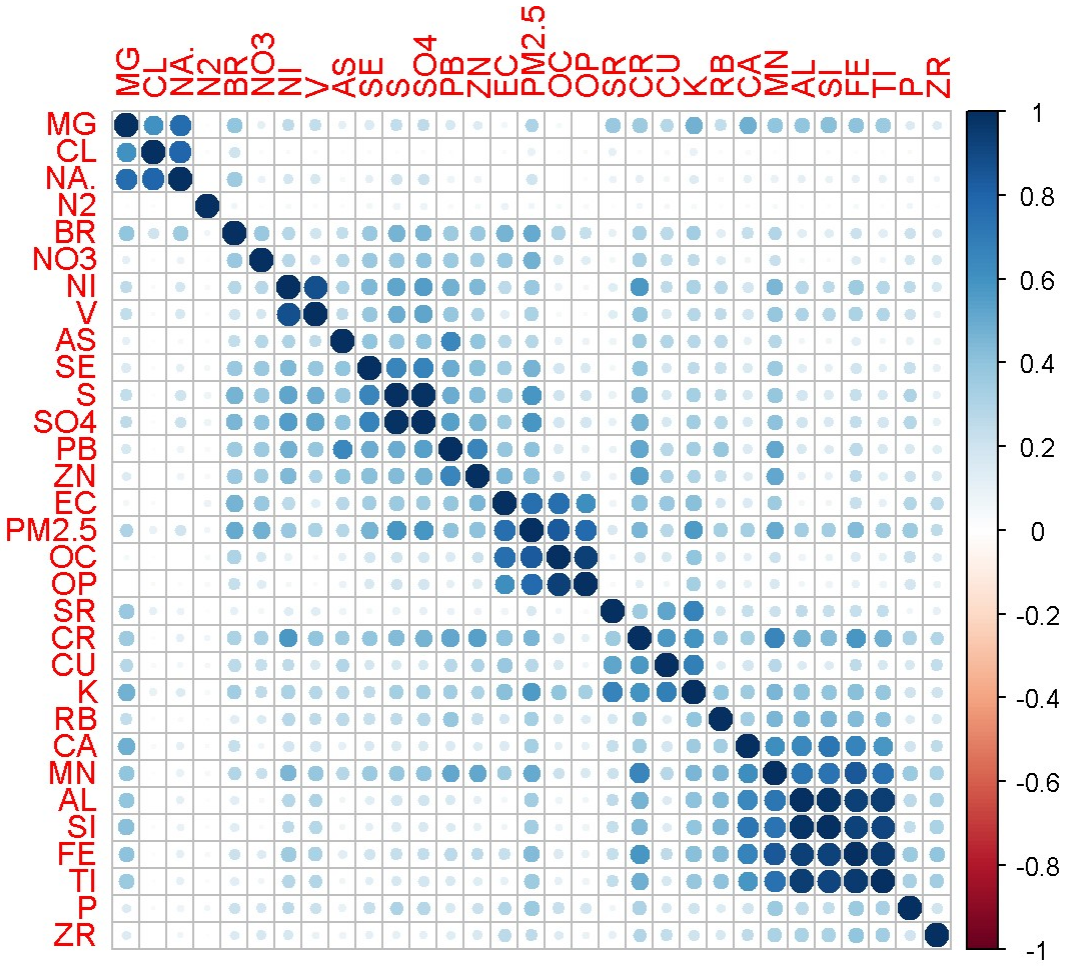


Figure (2.1) A2.2a: Aide-by-side Boxplots for fAbs and EC

```
R<-cor(US_DATA_LRG %>% dplyr::select(!all_of(c("SiteCode","Date","PM2.5_UNC"))))
corrplot(R,order="hclust")
```



— Step 2: Data prep for GMMs with mclust —

```
## --- Normalize US data by PM2.5 conc --
US_DATA_LRG_PM_norm<-US_DATA_LRG %>% dplyr::select(!c(PM2.5_UNC,SiteCode,Date)) %>%
mutate(across(everything()),./ PM2.5)
## --- Save factor to merge back into DF ---
US_DATA_LRG_factors<-US_DATA_LRG %>% dplyr::select(c(SiteCode,Date,PM2.5_UNC))

## --- Do the same for the test set ---
US_DATA_LRG_PM_norm_test<-US_DATA_LRG_test %>%
dplyr::select(!c(SiteCode,Date,PM2.5_UNC)) %>% transmute(across(everything()),./ PM
2.5)

## --- Save factor to merge back into DF ---
US_DATA_LRG_test_factors<-US_DATA_LRG_test %>% dplyr::select(c(SiteCode,Date,PM2.5_
UNC))

## --- FInal output (I'm sure there's a cleaner way to do this) ---
US_DATA_LRG_PM_norm1<-bind_cols(US_DATA_LRG_factors,US_DATA_LRG_PM_norm)
US_DATA_LRG_PM_norm_test1<-bind_cols(US_DATA_LRG_test_factors,US_DATA_LRG_PM_norm_t
est)

## --- Remove bad division by PM2.5 ---
logic_complete<-complete.cases(US_DATA_LRG_PM_norm1)
logic_complete_test<-complete.cases(US_DATA_LRG_PM_norm_test1)
US_DATA_LRG_PM_norm<-US_DATA_LRG_PM_norm1[complete.cases(US_DATA_LRG_PM_norm1),]
US_DATA_LRG_PM_norm_test<-US_DATA_LRG_PM_norm_test1[complete.cases(US_DATA_LRG_PM_n
orm_test1),]
```

```
any(is.na(US_DATA_LRG_PM_norm))
```

```
## [1] FALSE
```

```
any(is.na(US_DATA_LRG_PM_norm_test))
```

```
## [1] FALSE
```

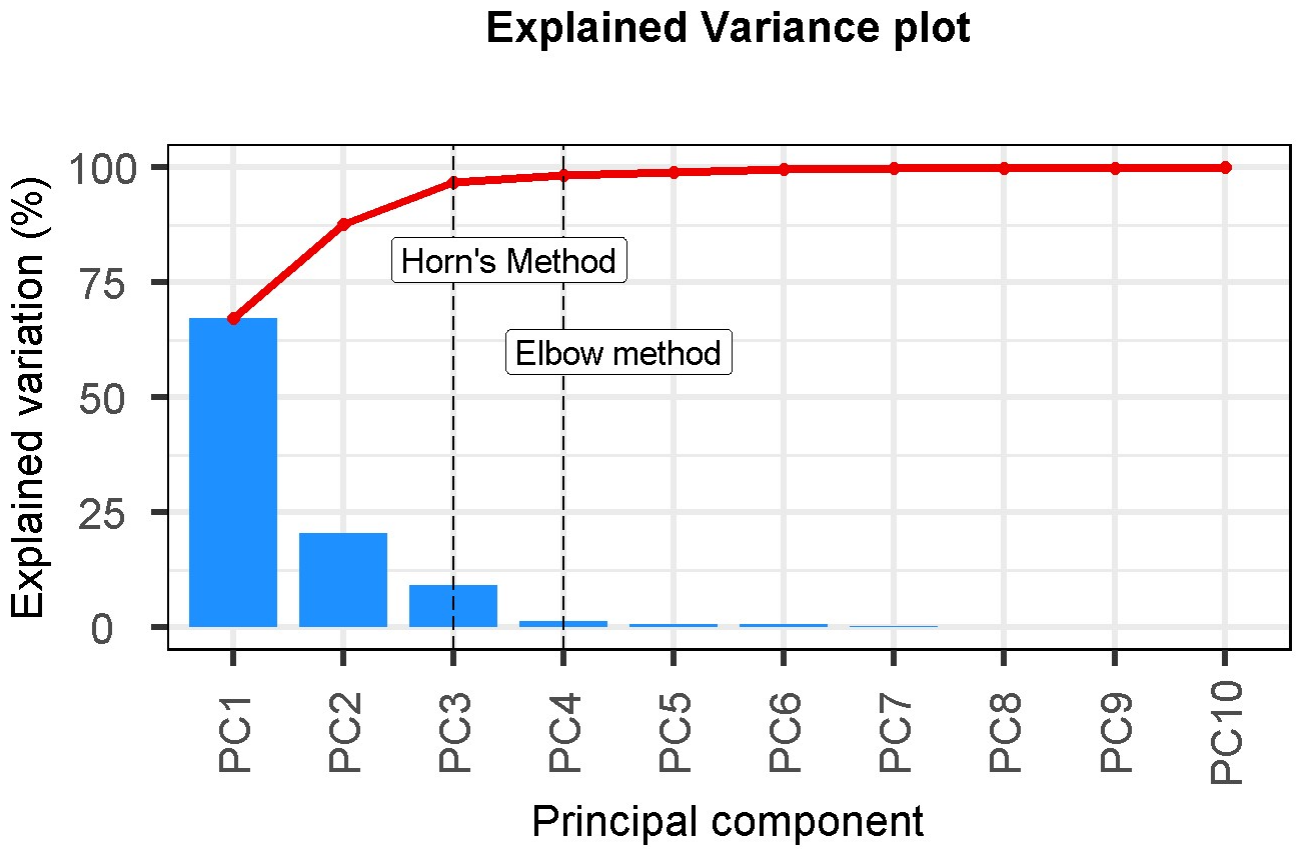
```
## --- Need to preprocess with PCA as these data are too large (and EM alg. will pr
obs. lead to non-convergance for high cluster #---
## ----
US_PCA_DATA_slim<-as_tibble(dplyr::select(US_DATA_LRG,!contains(c("SiteCode","Dat
e","PM2.5","PM2.5_UNC"))))
US_PCA_DATA_slim_test<-as_tibble(dplyr::select(US_DATA_LRG_test,!contains(c("SiteCo
de","Date","PM2.5","PM2.5_UNC"))))
### --- log transform ---

##Go through each row and determine if a value is zero
#row_sub = apply(US_PCA_DATA_slim, 1, function(row) all(row > 0))
#log_US_PCA_DATA_slim<-log(US_PCA_DATA_slim[row_sub,])

##Subset as usual
#log_US_PCA_DATA_slim<-log_US_PCA_DATA_slim[row_sub,]

### --- PCA with PCAtools package ---
# Damn! It does a transposed form of PCA bleh ---
US_PCA<-pca(US_PCA_DATA_slim,transposed = TRUE)

## --- Find elbow point on screeplot ---
elbow <- findElbowPoint(US_PCA$variance)
elbow
```

```
## PC4
##   4
```

```
horn <- parallelPCA(US_PCA_DATA_slim)
horn$n
```

```
## [1] 3
```

```
## --- Screeplot ---
  PCAtools::screeplot(US_PCA,
    components = getComponents(US_PCA, 1:10),vline = c(horn$n, elbow))+ggtitle("Exp
lained Variance plot")+theme(plot.title = element_text(hjust=0.5))+
    geom_label(aes(x = horn$n +0.5, y = 75,
      label = 'Horn\'s Method', vjust = 0, size = 5)) +
    geom_label(aes(x = elbow + 0.5, y = 55,
      label = 'Elbow method', vjust = 0, size = 5))
```

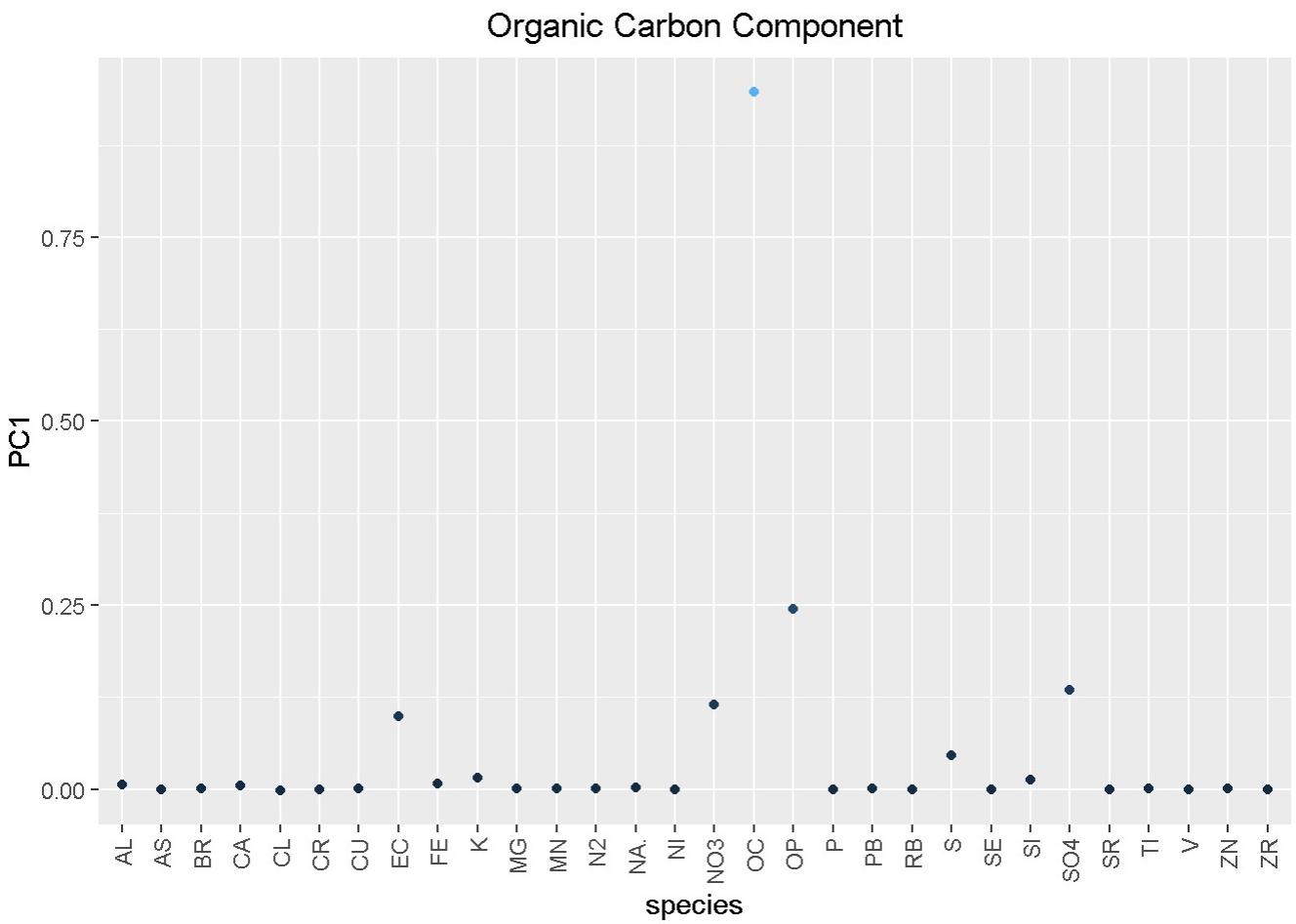



```
## --- Extract scores ---
scores<-as_tibble(US_PCA$rotated)
#names(scores)[31] <- "SiteCode"
## --- Extract scores and add to main data frame ---
US_DATA_w_scores<-add_column(US_DATA_LRG,scores)
## --- Extract loadings (format as tibble)---
loadings<-as_tibble(US_PCA$loadings,rownames="species")
loadings
```

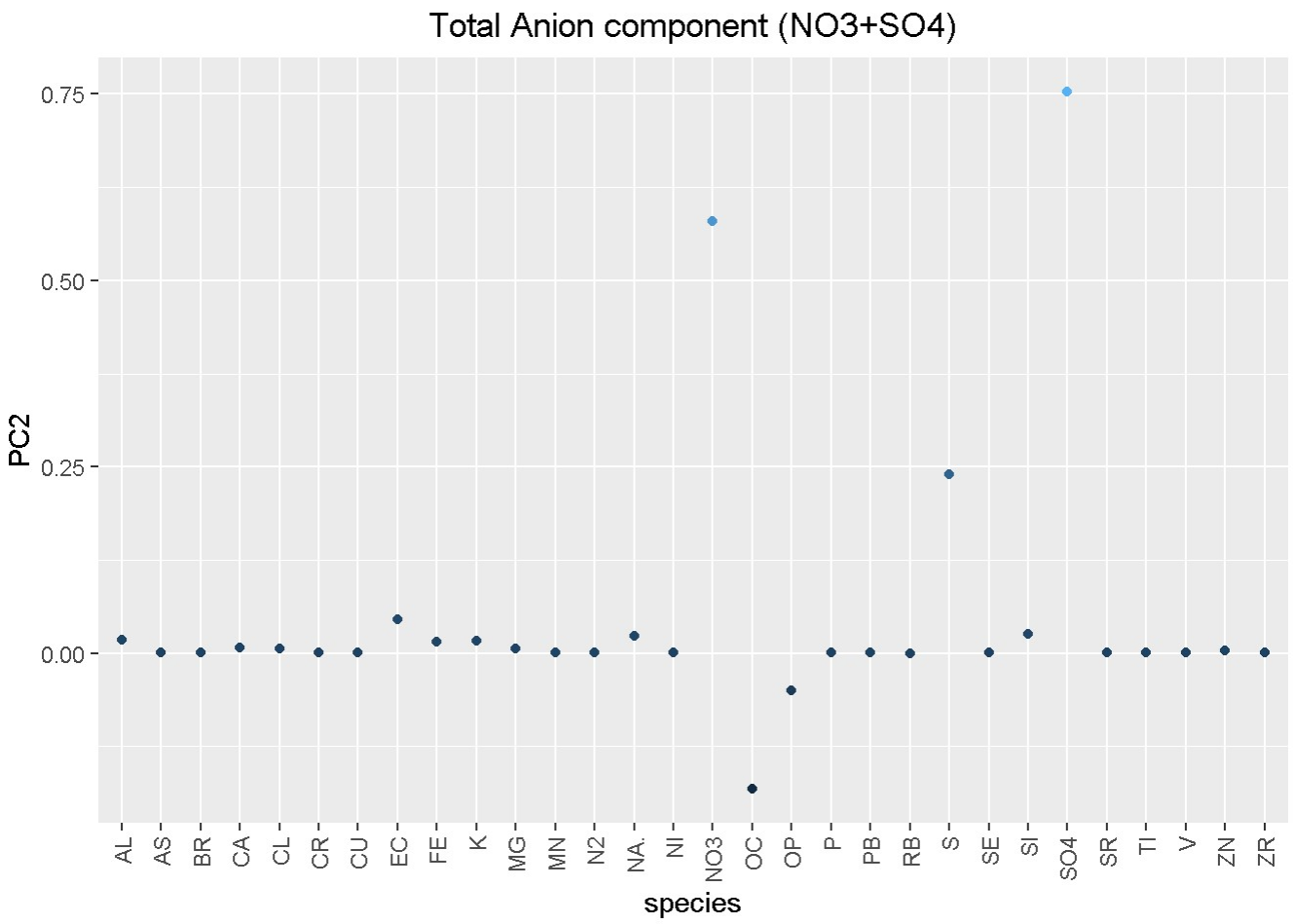
```
## # A tibble: 30 x 31
##    species      PC1      PC2      PC3      PC4      PC5      PC6      PC7
##    <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
##  1 EC       9.85e-2  4.47e-2 -2.51e-2  1.33e-2  5.36e-1  2.12e-1 -7.98e-1
##  2 OC       9.47e-1 -1.82e-1  2.54e-3 -3.71e-3  1.59e-1  2.13e-3  2.11e-1
##  3 OP       2.45e-1 -5.02e-2  3.16e-2  5.45e-2 -8.00e-1 -9.65e-2 -5.27e-1
##  4 AL       6.32e-3  1.77e-2  2.57e-2 -4.36e-1 -3.12e-2 -2.66e-3  4.45e-3
##  5 AS       5.71e-5  2.65e-4  4.77e-5  3.84e-6  3.21e-4  1.51e-4 -6.52e-4
##  6 BR       3.14e-4  6.72e-4  2.69e-5 -3.91e-4  2.04e-3 -1.71e-3 -1.69e-3
##  7 CA       4.39e-3  6.96e-3  7.73e-3 -1.58e-1  9.22e-3 -7.68e-3 -6.37e-2
##  8 CL      -1.33e-3  5.86e-3 -5.81e-4  3.50e-3  1.48e-1 -7.68e-1 -1.68e-1
##  9 CR       3.50e-5  1.17e-4  2.48e-5 -3.92e-4  3.25e-4  7.81e-5 -3.39e-4
## 10 CU       2.52e-4  4.83e-4 -3.03e-4 -1.33e-3  4.11e-3  5.71e-4 -3.66e-3
## # ... with 20 more rows, and 23 more variables: PC8 <dbl>, PC9 <dbl>,
## #   PC10 <dbl>, PC11 <dbl>, PC12 <dbl>, PC13 <dbl>, PC14 <dbl>, PC15 <dbl>,
## #   PC16 <dbl>, PC17 <dbl>, PC18 <dbl>, PC19 <dbl>, PC20 <dbl>, PC21 <dbl>,
## #   PC22 <dbl>, PC23 <dbl>, PC24 <dbl>, PC25 <dbl>, PC26 <dbl>, PC27 <dbl>,
## #   PC28 <dbl>, PC29 <dbl>, PC30 <dbl>
```

```
## --- Project test samples onto principal components ---
## --- Reformat for use with predict.prcomp
  US_PCA.prcomp <- list(sdev = US_PCA$sdev,
    rotation = data.matrix(US_PCA$loadings),
    x = data.matrix(US_PCA$rotated),
    center = TRUE, scale = FALSE)

  class(US_PCA.prcomp) <- 'prcomp'
## -- Estimate test set scores ---
scores_test<-as_tibble(predict(US_PCA.prcomp, newdata = US_PCA_DATA_slim_test))
```

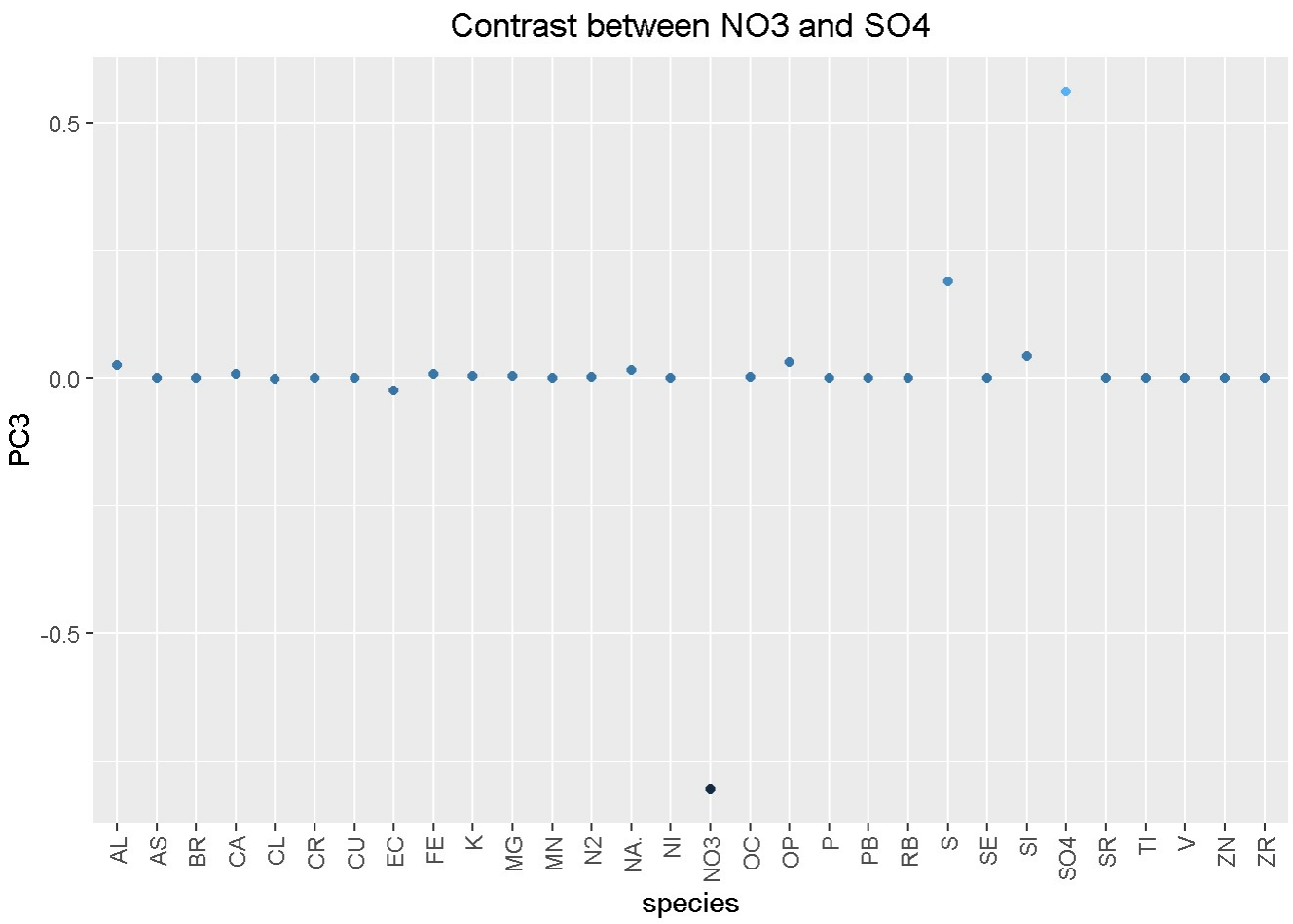
```
## --- PC1 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC1,color=PC1))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Organic Carbon Component")+theme(plot.title = element_text(hj
ust = 0.5))
```



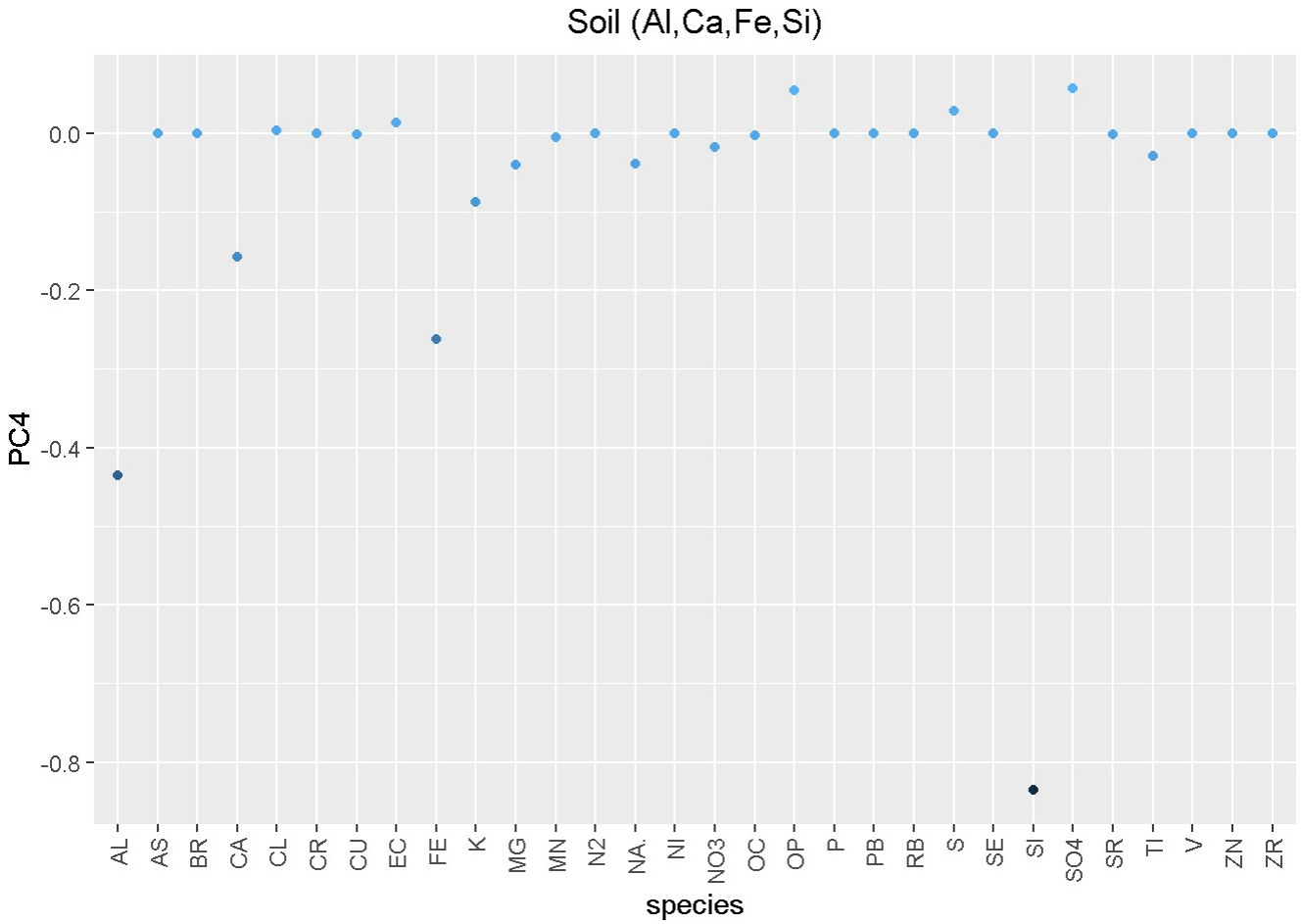

```
## --- PC2 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC2,color=PC2))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Total Anion component (NO3+SO4)")+theme(plot.title = element_
text(hjust = 0.5))
```



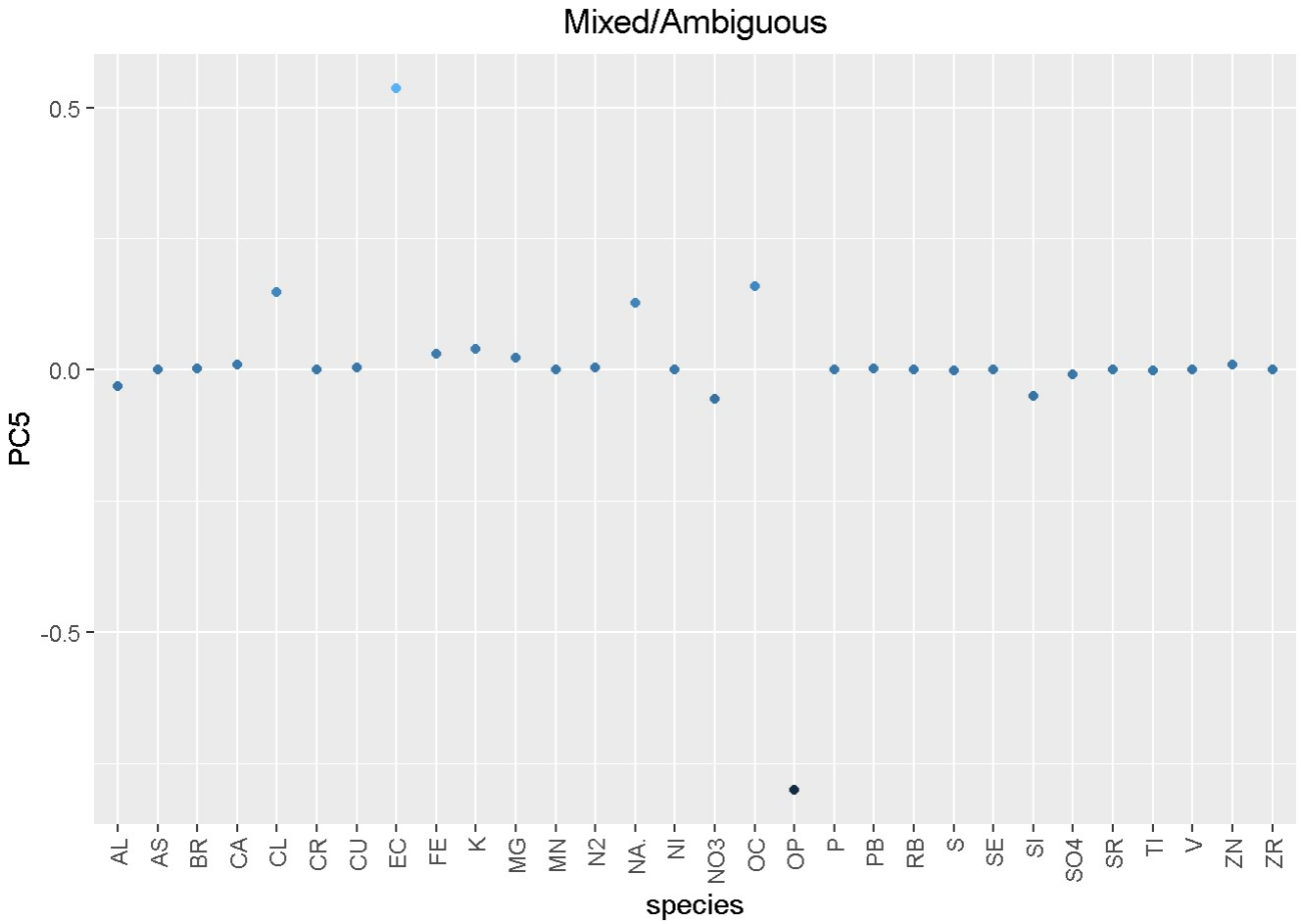

```
## --- PC3 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC3,color=PC3))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Contrast between NO3 and SO4")+theme(plot.title = element_tex
t(hjust = 0.5))
```



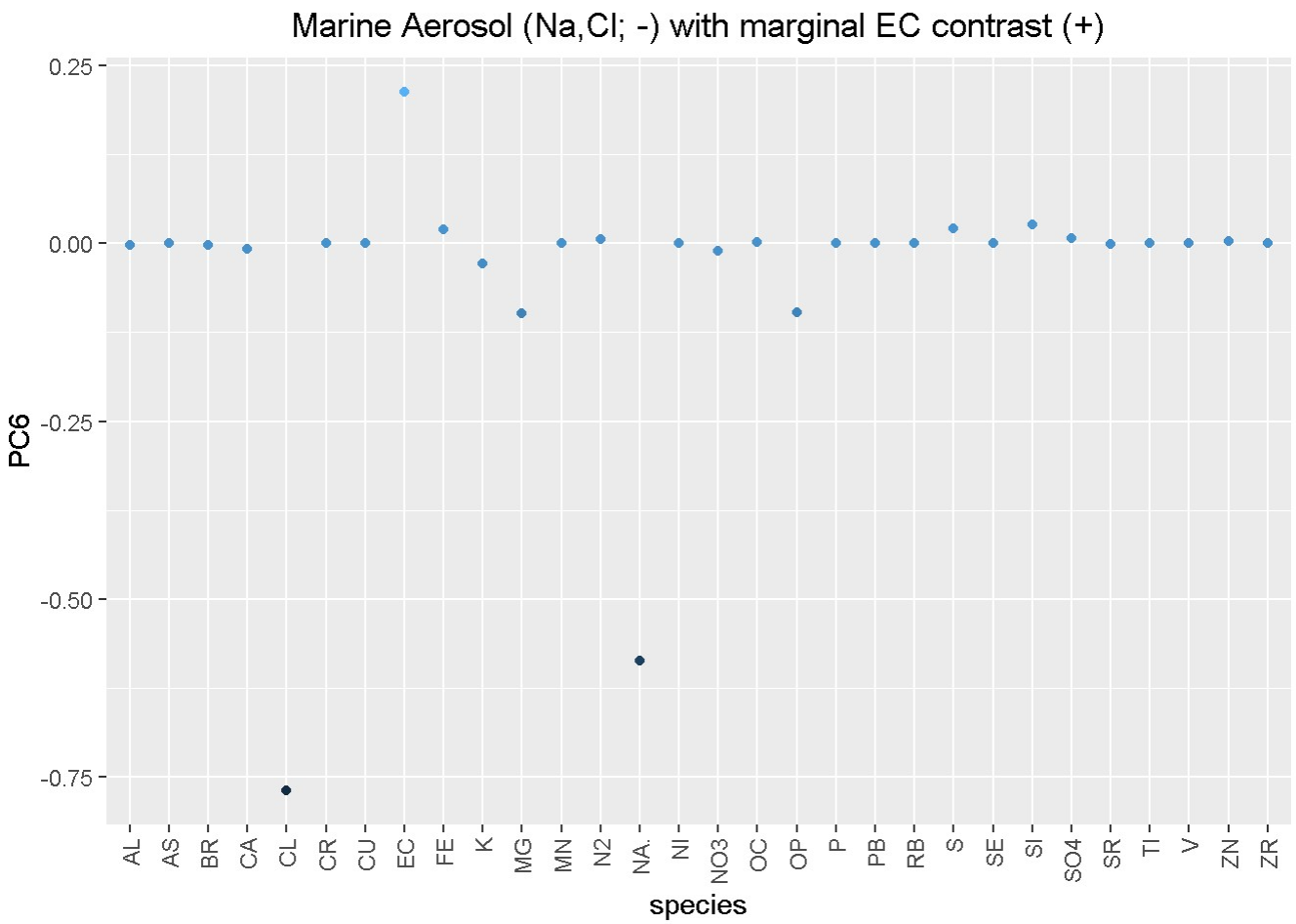

```
## --- PC4 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC4,color=PC4))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Soil (Al,Ca,Fe,Si)")+theme(plot.title = element_text(hjust =
0.5))
```




```
## --- PC5 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC5,color=PC5))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Mixed/Ambiguous")+theme(plot.title = element_text(hjust = 0.
5))
```



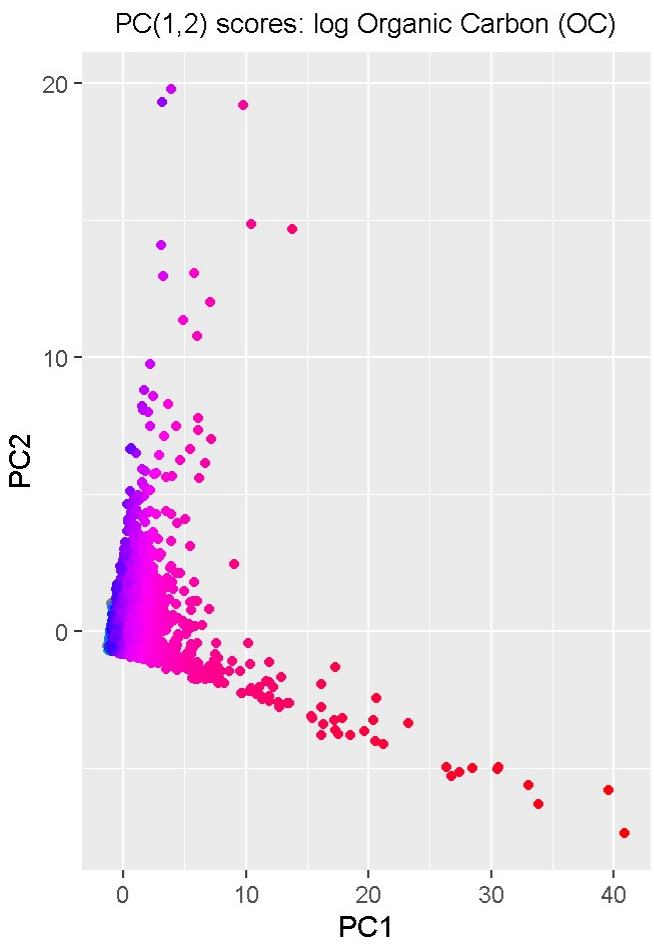

```
## --- PC6 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC6,color=PC6))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Marine Aerosol (Na,Cl; -) with marginal EC contrast (+)")+the
me(plot.title = element_text(hjust = 0.5))
```

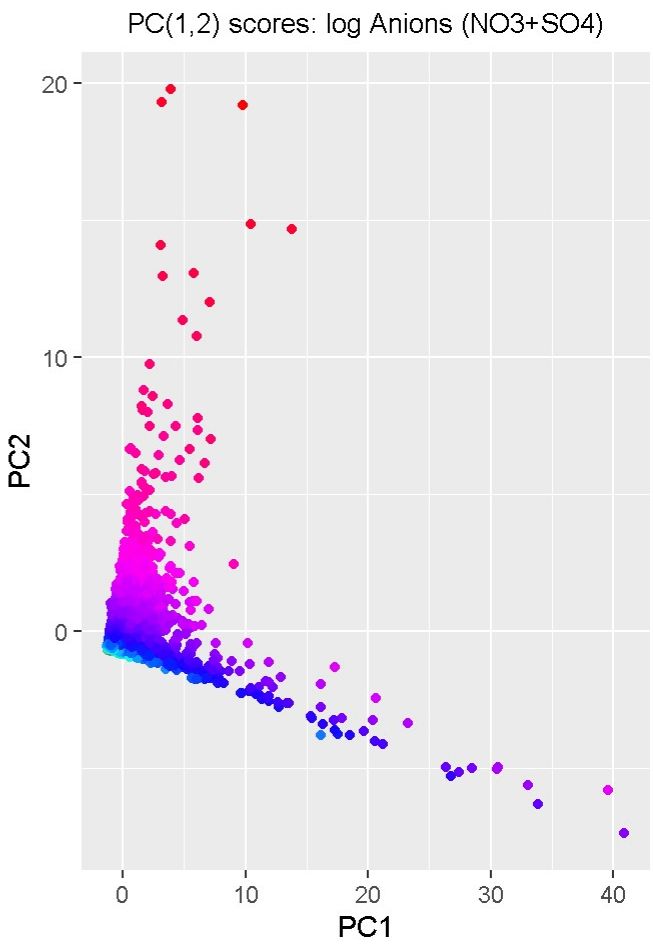



```
P2<-ggplot(data =US_DATA_w_scores,aes(x = PC1, y = PC2)) +
    geom_point(mapping = aes(color = log(SO4+NO3)))+theme(plot.title = element_text
(hjust = 0.5,size=10),legend.position = "none")+ scale_color_gradientn(colours = ra
inbow(20,start=0.25,end=1))+ggtitle("PC(1,2) scores: log Anions (NO3+SO4)")
P1<-ggplot(data =US_DATA_w_scores,aes(x = PC1, y = PC2)) +
    geom_point(mapping = aes(color = log(OC)))+theme(plot.title = element_text(hjus
t = 0.5,size=10),legend.position = "none")+ scale_color_gradientn(colours = rainbow
(20,start=0.25,end=1))+ggtitle("PC(1,2) scores: log Organic Carbon (OC)")

grid.arrange(P1,P2,nrow=1)
```

```
## Warning in log(OC): NaNs produced
```

```
## Warning in log(SO4 + NO3): NaNs produced
```

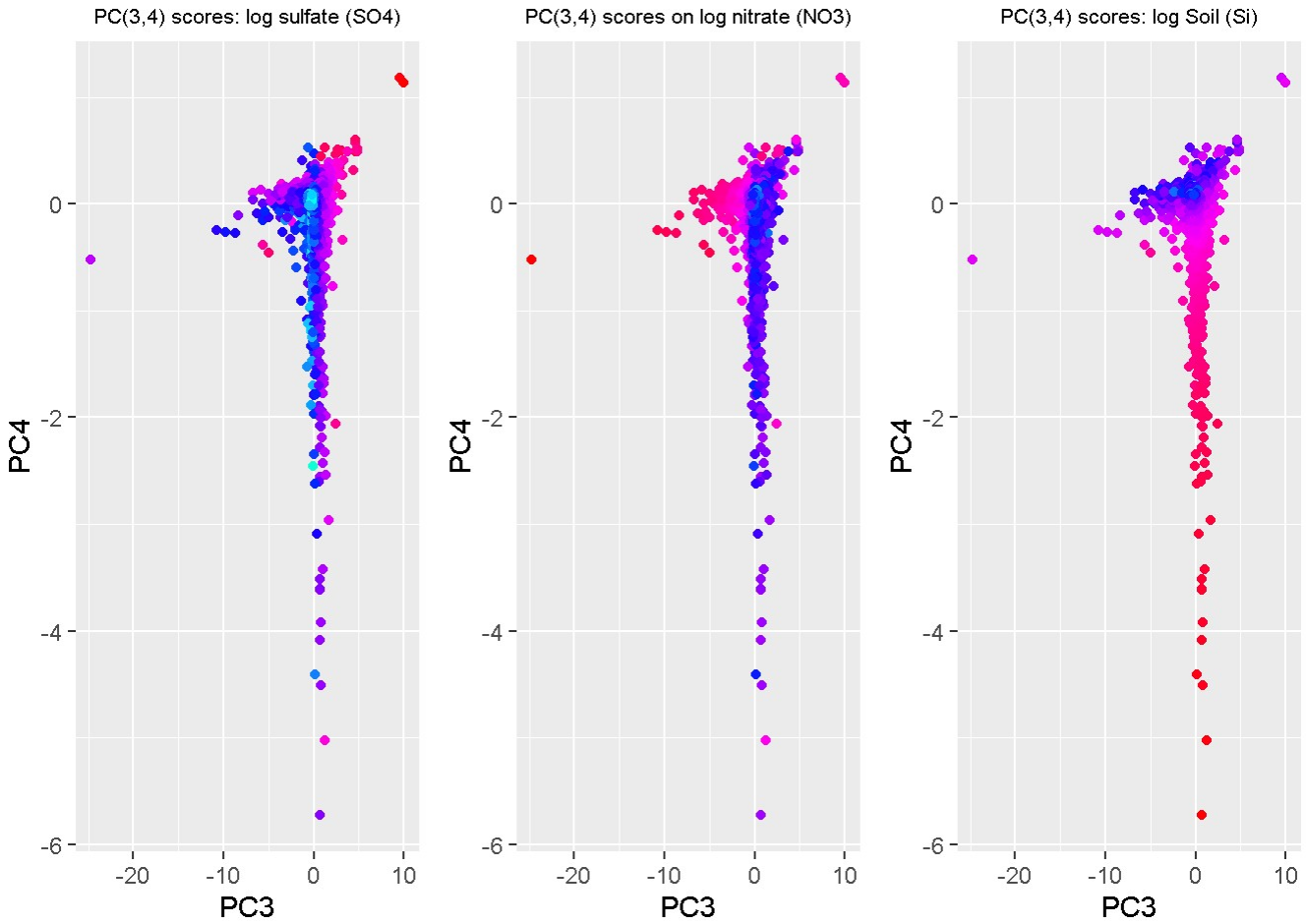

PC(1,2) scores: log Organic Carbon (OC)
PC2
PC1
20
10
0
0
10
20
30
40


PC(1,2) scores: log Anions (NO3+SO4)
PC2
PC1
20
10
0
0
10
20
30
40

```
P3<-ggplot(data =US_DATA_w_scores,aes(x = PC3, y = PC4)) +
    geom_point(mapping = aes(color = log(SO4)))+theme(plot.title = element_text(hju
st = 0.5,size=8),legend.position = "none")+ scale_color_gradientn(colours = rainbow
(20,start=0.25,end=1))+ggtitle("PC(3,4) scores: log sulfate (SO4)")

P4<-ggplot(data =US_DATA_w_scores,aes(x = PC3, y = PC4)) +
    geom_point(mapping = aes(color = log(NO3)))+theme(plot.title = element_text(hju
st = 0.5,size=8),legend.position = "none")+ scale_color_gradientn(colours = rainbow
(20,start=0.25,end=1))+ggtitle("PC(3,4) scores on log nitrate (NO3)")

# --- IMPROVE Soil equation ---
# --- Attests to the general validity of the soil equation ---
# SOIL Eqn = 2.20*Al + 2.49*Si + 1.63*Ca + 2.42*Fe + 1.94*Ti

P5<-ggplot(data =US_DATA_w_scores,aes(x = PC3, y = PC4)) +
    geom_point(mapping = aes(color = log(2.2*AL+2.49*SI+1.63*CA+2.42*FE+1.94*TI)))+
theme(plot.title = element_text(hjust = 0.5,size=8),legend.position = "none")+ scal
e_color_gradientn(colours = rainbow(20,start=0.25,end=1))+ggtitle("PC(3,4) scores:
log Soil (Si)")

## --- Not the most efficient but whatevs ---
ind_grp <- US_DATA_w_scores %>% group_by(SiteCode) %>% group_indices
US_META_Slim<-US_META %>% filter(Code %in% US_DATA_w_scores$SiteCode)
EW<-rep(NA,length(US_DATA_w_scores$SiteCode))
for(k in 1:length(unique(US_DATA_w_scores$SiteCode))){
    EW[ind_grp==k]<-US_META_Slim$WE_US[k]
}
US_DATA_w_scores<-add_column(US_DATA_w_scores,EW_indicator=EW)
## --- East-West binary color coding ---
# --- Nopt informative
P6<-ggplot(data =US_DATA_w_scores,aes(x = PC3, y = PC4)) +
    geom_point(mapping = aes(color = EW))+
  theme(plot.title = element_text(hjust = 0.5,size=8),legend.position = "none")+
  ggtitle("PC(3,4) scores: East-West divide")

grid.arrange(P3,P4,P5,nrow=1)
```

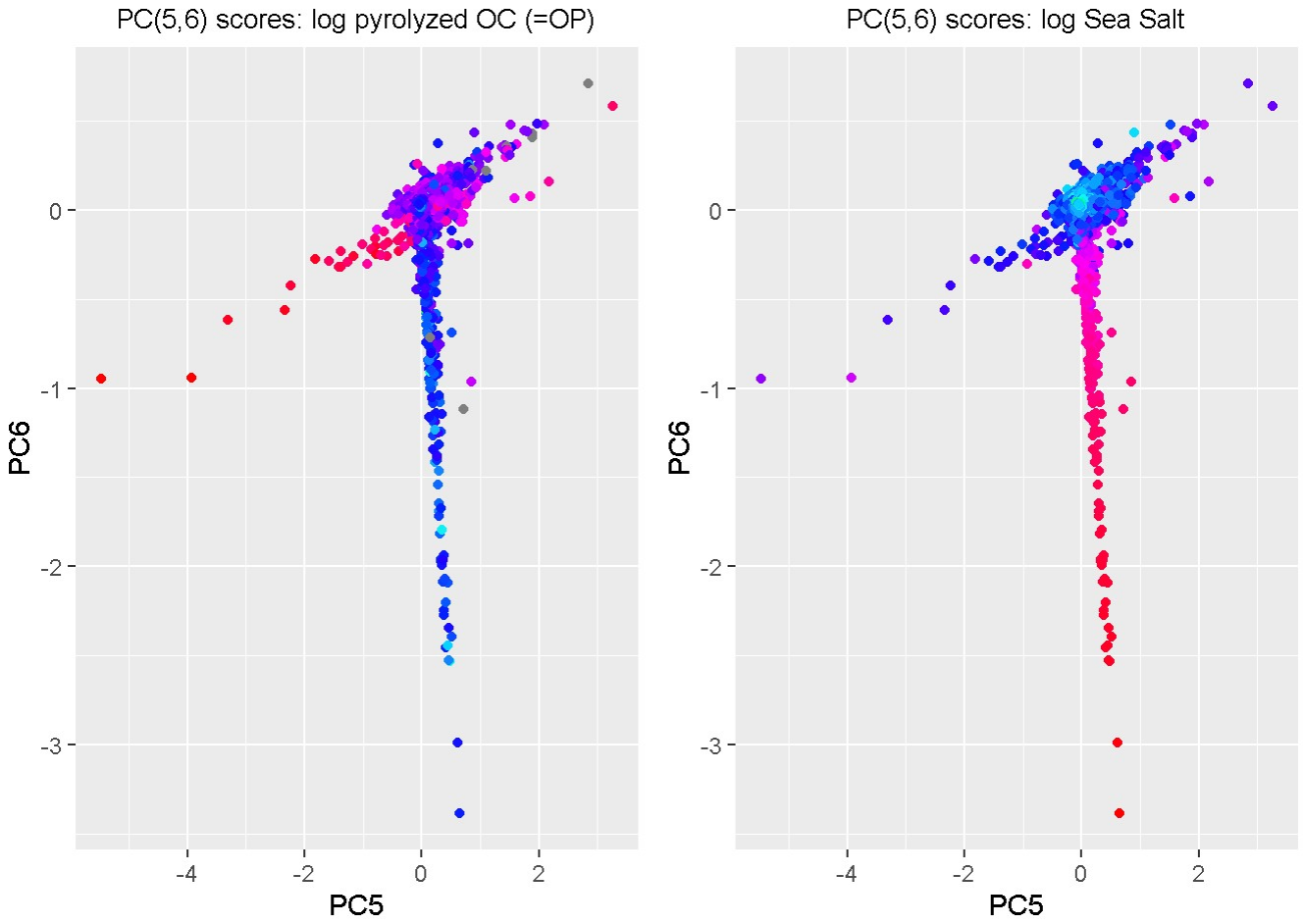
```
## Warning in log(NO3): NaNs produced
```

```
## Warning in log(2.2 * AL + 2.49 * SI + 1.63 * CA + 2.42 * FE + 1.94 * TI): NaNs
## produced
```




```
# --- Total carbon: TC = OC + EC---
P6<-ggplot(data =US_DATA_w_scores,aes(x = PC5, y = PC6)) +
    geom_point(mapping = aes(color = log(OC+EC)))+theme(plot.title = element_text(h
just = 0.5,size=8),legend.position = "none")+ scale_color_gradientn(colours = rainb
ow(20,start=0.25,end=1))+ggtitle("PC(5,6) scores: log Carbon (OC+EC)")
# --- Pyrolyzed OC (=OP) ---
P7<-ggplot(data =US_DATA_w_scores,aes(x = PC5, y = PC6)) +
    geom_point(mapping = aes(color = log(OP)))+theme(plot.title = element_text(hjus
t = 0.5,size=10),legend.position = "none")+ scale_color_gradientn(colours = rainbow
(20,start=0.25,end=1))+ggtitle("PC(5,6) scores: log pyrolyzed OC (=OP)")
# --- IMPROVE Eqn for Marine Aerosol: 1.8*Cl

P8<-ggplot(data =US_DATA_w_scores,aes(x = PC5, y = PC6)) +
    geom_point(mapping = aes(color = log(1.8*CL)))+theme(plot.title = element_text
(hjust = 0.5,size=10),legend.position = "none")+ scale_color_gradientn(colours = ra
inbow(20,start=0.25,end=1))+ggtitle("PC(5,6) scores: log Sea Salt")

grid.arrange(P7,P8,nrow=1)
```

```
## Warning in log(1.8 * CL): NaNs produced
```



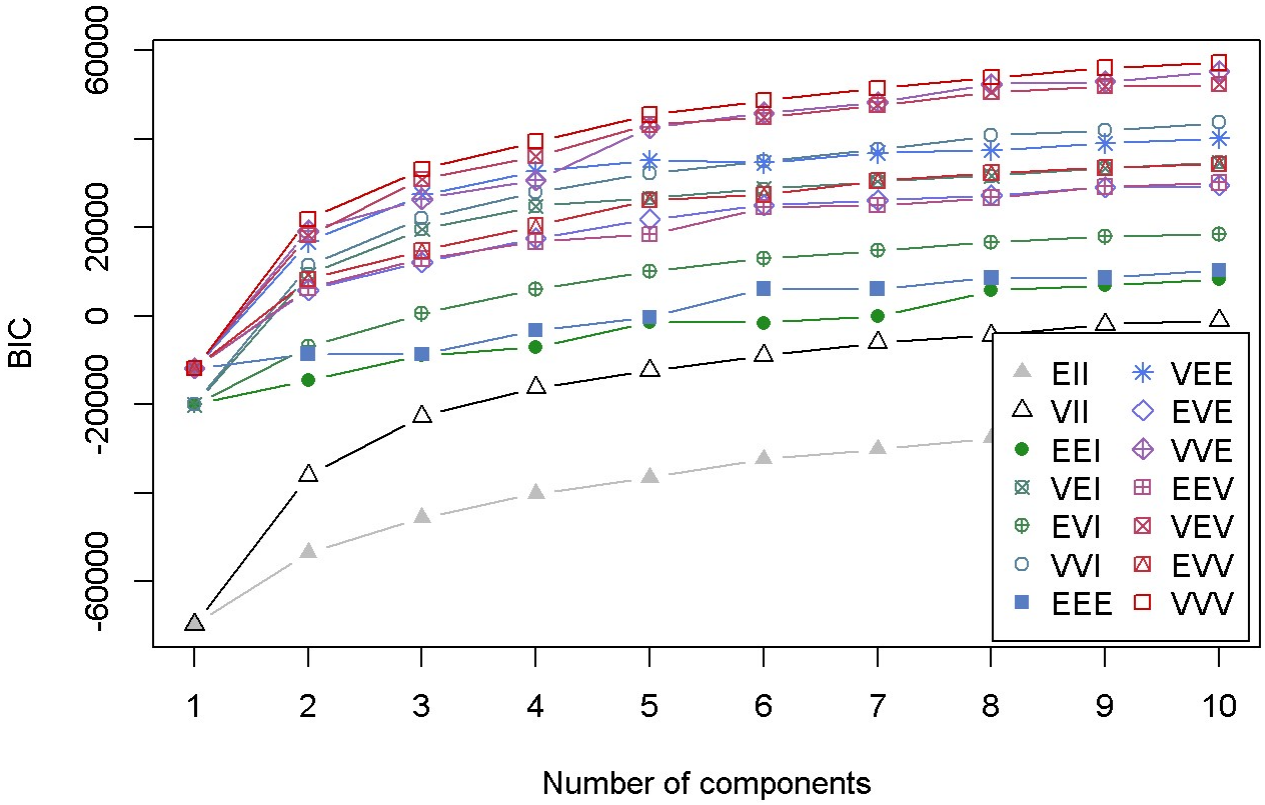
— Step 3: Gaussian Mixture Models: clustering on components —

```
# --- Step 3.1: Number of PCs to consider ---
num_PCs<-6
num_clust<-10 #Interest of simplicity
# --- Step 2: Select scores from US_DATA_scores structure ---
GMM_scores<-US_DATA_w_scores %>% dplyr::select(num_range(prefix="PC",range=1:num_PC
s))
```

```
# --- Step 3.1: GMM mixture model initialization with k-means---
k_clust=100
kmeans_partition<-kmeans(GMM_scores, k_clust, iter.max = 100, nstart = 1)

# --- Step 3.2: Further Initialization with HCA ---
hc_out<-hc(GMM_scores,partition = kmeans_partition$cluster,minclus=1,hcUse="VARS")
#--- Don't quite see how to use this yet... This is a bit buggy

# --- Step 3.3: Option to specify noise... That's interesting
# --- We have uncertainties for PM2.5: therefore we can estimate noise as samples < minum detection limit where MDL ~= 3*min(UNC)
PM_noise<-which(US_DATA_LRG$PM2.5 <3*min(US_DATA_LRG$PM2.5_UNC))
# --- PErhaps we can make a good signal-to-noise ratio argument ---
SNR<-US_DATA_LRG %>% dplyr::select(PM2.5,PM2.5_UNC,SiteCode,Date)%>% mutate(SNR_PM = PM2.5/PM2.5_UNC,Date =as.Date(Date,"%m/%d/%Y"))

SNR_sort<-arrange(SNR,Date)

#qplot(x=as.factor(Date),y=SNR_PM,data=SNR_sort,geom="boxplot",
#  main = "Signal-to-noise ratio: PM2.5",xlab="Date",ylab="SNR")+theme(plot.title=element_text(hjust = 0.5))+ geom_smooth(method= "loess",span=0.1 ,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))


## --- Define "noise" for GMM as sample with low SNR and inordinately high ---> separate horses from zebra's given lognormality
PM_noise<-which(SNR$SNR_PM <quantile(SNR$SNR_PM,0.05) | SNR$SNR_PM > quantile(SNR$SNR_PM,0.95))
```

```
# --- Step 3.4: Run GMM with initial paramters ---
init_list<-list(hcPairs=NULL,noise=PM_noise)
# --- It'd be nice if I could use noise to
#GMM_BIC<-mclustBIC(GMM_scores,G=1:num_clust,initialization = init_list)

load("GMM_BIC_10_clust_init_SNR.RData")
#save("GMM_BIC",file="GMM_BIC_10_clust_init_SNR.RData")
```

```
plot(GMM_BIC)
```




```
summary(GMM_BIC)
```

```
## Best BIC values:
##             VVV,10     VVV,9    VVE,10
## BIC      57287.91 56118.809 55302.303
## BIC diff     0.00 -1169.099 -1985.605
```

```
BIC_best <- Mclust(GMM_scores, x = GMM_BIC)
summary(BIC_best, parameters = TRUE)
```

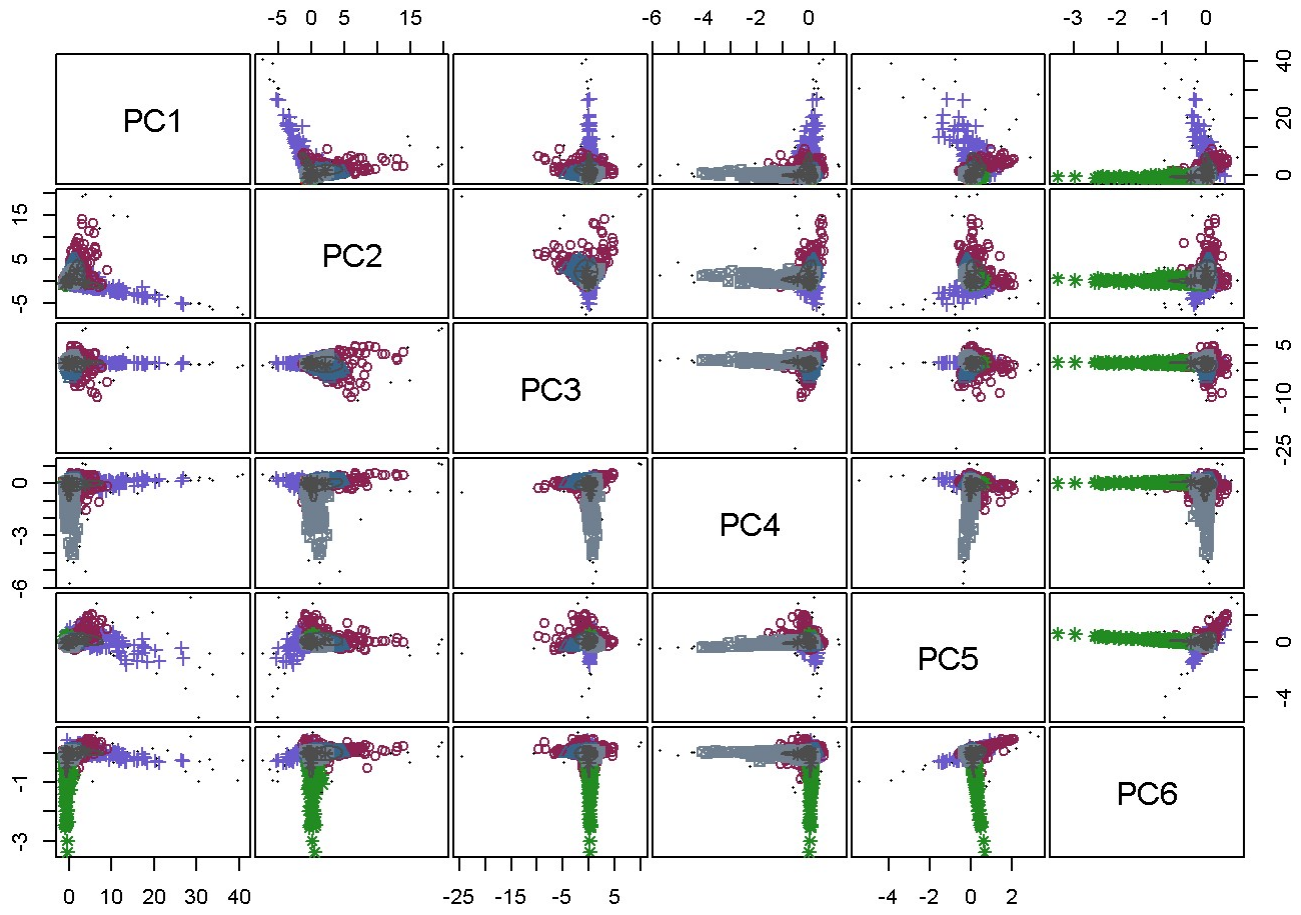
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 10
## components and a noise term:
##
##  log-likelihood    n  df      BIC      ICL
##        29917.58 8647 281 57287.91 54399.16
##
## Clustering table:
##    1    2    3    4    5    6    7    8    9   10    0
## 1176 1703 1114  353 1158  894  255  648  671  640   35
##
## Mixing probabilities:
##          1          2          3          4          5          6          7
## 0.13591772 0.19279098 0.12440110 0.04135019 0.12768836 0.10496367 0.03046932
##          8          9         10          0
## 0.07405139 0.08358606 0.08061601 0.00416521
##
## Means:
##           [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
## PC1 0.69267678 -0.53172131 -0.94673005  3.73514033 -0.58903317 -0.36710042
## PC2 0.11473832 -0.27470155 -0.54359346 -0.47898731 -0.46449393  0.07355864
## PC3 0.40825407  0.04306974 -0.12812754  0.06864350 -0.08562184 -0.07782542
## PC4 0.07223202 -0.04053731  0.06254042 -0.02889360  0.07232752  0.06571892
## PC5 0.03869801 -0.03573801 -0.03561255  0.21450374 -0.02073530 -0.05305863
## PC6 0.04800211  0.02748356  0.03495178  0.04560729  0.03896338  0.04413208
##            [,7]        [,8]        [,9]        [,10]
## PC1  1.87661638 -0.55441969  0.16601632 -0.006328252
## PC2  2.22410794 -0.04802350  0.96283811  0.290698525
## PC3 -0.50177185 -0.02740213 -0.41068581  0.285816037
## PC4 -0.05194260  0.07980363  0.09836590 -0.424493963
## PC5  0.25803170  0.06032060 -0.05196334 -0.018771251
## PC6 -0.01199697 -0.34667240  0.01684476 -0.028016497
##
## Variances:
## [,,1]
##             PC1          PC2          PC3           PC4          PC5
## PC1 0.661631946  0.002940929  0.104408793  0.0147379892  0.028254138
## PC2 0.002940929  0.542646376  0.366210349  0.0399162678 -0.042831036
## PC3 0.104408793  0.366210349  0.276786926  0.0304875420 -0.026871613
## PC4 0.014737989  0.039916268  0.030487542  0.0093530151 -0.002494648
## PC5 0.028254138 -0.042831036 -0.026871613 -0.0024946483  0.012570132
## PC6 0.006424865  0.005879452  0.005359355  0.0009828385  0.001389437
##              PC6
## PC1 0.0064248653
## PC2 0.0058794518
## PC3 0.0053593555
## PC4 0.0009828385
```

```
## PC5 0.0013894371
## PC6 0.0010906003
## [,,2]
##                PC1           PC2           PC3           PC4           PC5
## PC1  0.1215545948  0.0164612544  0.0259295301 -0.0007879201  0.0044774769
## PC2  0.0164612544  0.0563012962  0.0353713357 -0.0003073212 -0.0036470039
## PC3  0.0259295301  0.0353713357  0.0274587674  0.0014854915 -0.0016569797
## PC4 -0.0007879201 -0.0003073212  0.0014854915  0.0150054870  0.0005786928
## PC5  0.0044774769 -0.0036470039 -0.0016569797  0.0005786928  0.0012748550
## PC6  0.0014898853 -0.0002220515  0.0004493597 -0.0002623696  0.0001638180
##                PC6
## PC1  0.0014898853
## PC2 -0.0002220515
## PC3  0.0004493597
## PC4 -0.0002623696
## PC5  0.0001638180
## PC6  0.0003151619
## [,,3]
##                PC1           PC2           PC3           PC4           PC5
## PC1  0.0296123446  3.885329e-03  5.512054e-03 -1.924867e-03  1.045736e-03
## PC2  0.0038853292  6.500421e-03  4.115279e-03 -1.408572e-03 -5.094681e-04
## PC3  0.0055120537  4.115279e-03  3.279830e-03 -9.242974e-04 -1.725490e-04
## PC4 -0.0019248669 -1.408572e-03 -9.242974e-04  1.228676e-03  8.992743e-05
## PC5  0.0010457364 -5.094681e-04 -1.725490e-04  8.992743e-05  2.691386e-04
## PC6  0.0002418708  7.974206e-05  8.206355e-05 -1.222439e-05  3.593502e-05
##                PC6
## PC1  2.418708e-04
## PC2  7.974206e-05
## PC3  8.206355e-05
## PC4 -1.222439e-05
## PC5  3.593502e-05
## PC6  2.317346e-05
## [,,4]
##             PC1         PC2          PC3          PC4         PC5          PC6
## PC1 18.2636863 -3.77536332  0.115398611  0.236821429 -0.72762885 -0.244632991
## PC2 -3.7753633  1.17476102  0.164350436 -0.022027223  0.10431514  0.049508990
## PC3  0.1153986  0.16435044  0.188958361  0.017422409 -0.02247076  0.001827713
## PC4  0.2368214 -0.02202722  0.017422409  0.028393586 -0.01027786 -0.001461844
## PC5 -0.7276288  0.10431514 -0.022470764 -0.010277861  0.16123886  0.032626210
## PC6 -0.2446330  0.04950899  0.001827713 -0.001461844  0.03262621  0.010754828
## [,,5]
##                PC1           PC2           PC3           PC4           PC5
## PC1  0.1102659756 -1.059609e-02  1.311907e-02 -7.522516e-04  0.0065640094
## PC2 -0.0105960868  2.037331e-02  9.560561e-03  1.401215e-04 -0.0024687423
## PC3  0.0131190742  9.560561e-03  1.099495e-02 -4.374116e-05 -0.0003379081
## PC4 -0.0007522516  1.401215e-04 -4.374116e-05  8.597653e-04 -0.0001094266
## PC5  0.0065640094 -2.468742e-03 -3.379081e-04 -1.094266e-04  0.0016234481
## PC6  0.0008893522  4.710733e-05  1.855381e-04 -6.968797e-06  0.0002865969
##                PC6
## PC1  8.893522e-04
## PC2  4.710733e-05
```

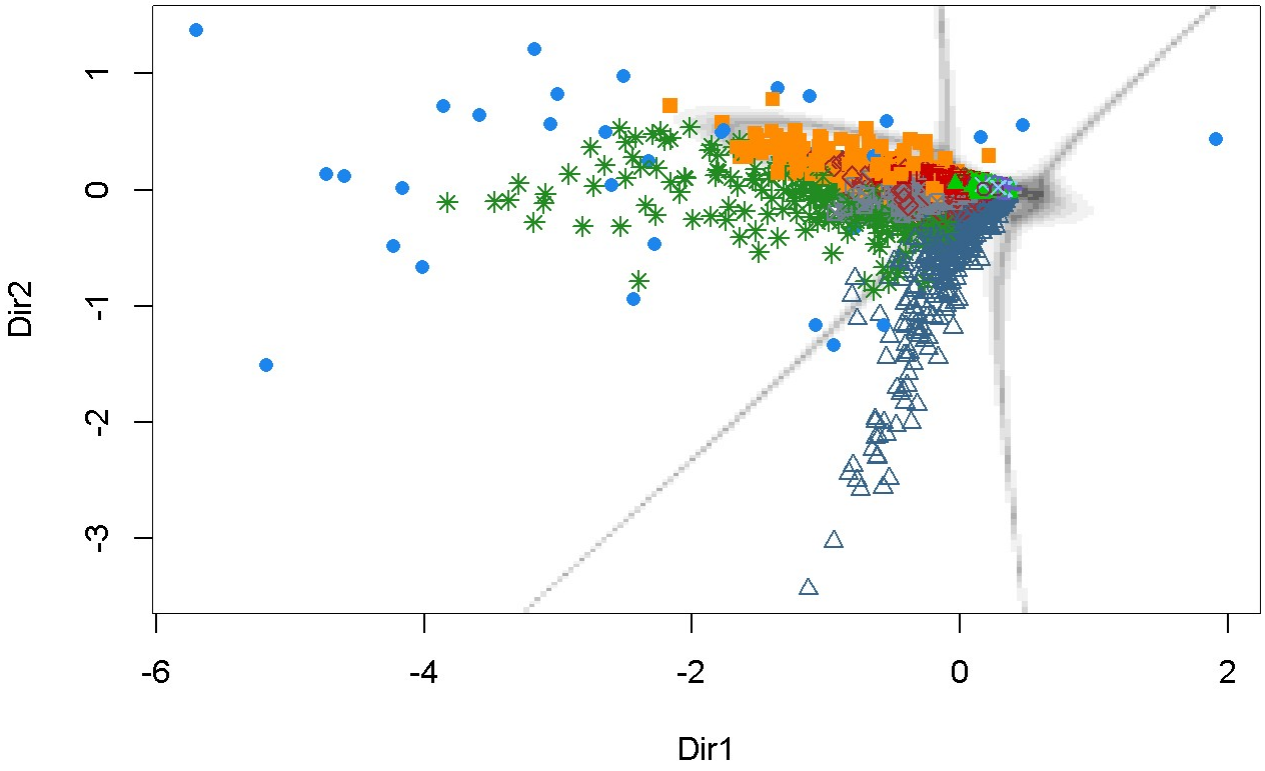
```
## PC3  1.855381e-04
## PC4 -6.968797e-06
## PC5  2.865969e-04
## PC6  9.083168e-05
## [,,6]
##              PC1           PC2          PC3           PC4           PC5
## PC1 0.145485565  0.0420897762 0.0438060883  0.0027585073  0.0038103592
## PC2 0.042089776  0.1197876730 0.0263081906  0.0061699570 -0.0067087859
## PC3 0.043806088  0.0263081906 0.1268234842  0.0074096324  0.0002473831
## PC4 0.002758507  0.0061699570 0.0074096324  0.0049818395 -0.0004175308
## PC5 0.003810359 -0.0067087859 0.0002473831 -0.0004175308  0.0017569941
## PC6 0.002329883  0.0009057461 0.0019901732  0.0001555978  0.0003025449
##              PC6
## PC1 0.0023298833
## PC2 0.0009057461
## PC3 0.0019901732
## PC4 0.0001555978
## PC5 0.0003025449
## PC6 0.0001459026
## [,,7]
##             PC1        PC2         PC3          PC4         PC5          PC6
## PC1  3.0208824  0.9416265 -0.72352449  0.007411600  0.50742774  0.246374154
## PC2  0.9416265  6.7183064  0.21024627  0.438679672 -0.38312279  0.124838793
## PC3 -0.7235245  0.2102463  4.78181203  0.145509138 -0.02381941 -0.041123501
## PC4  0.0074116  0.4386797  0.14550914  0.101340858 -0.03682091  0.004391653
## PC5  0.5074277 -0.3831228 -0.02381941 -0.036820914  0.21352379  0.050655184
## PC6  0.2463742  0.1248388 -0.04112350  0.004391653  0.05065518  0.058366431
## [,,8]
##               PC1          PC2          PC3           PC4           PC5
## PC1  0.1332829460  0.074166162  0.018231782  0.0002272106  0.0109193460
## PC2  0.0741661621  0.159126393  0.050304903  0.0010077052  0.0131403782
## PC3  0.0182317817  0.050304903  0.041724618  0.0025204957  0.0059746342
## PC4  0.0002272106  0.001007705  0.002520496  0.0009929167 -0.0009371342
## PC5  0.0109193460  0.013140378  0.005974634 -0.0009371342  0.0112231500
## PC6 -0.0031258580 -0.059439783 -0.030086322  0.0041913492 -0.0472726200
##              PC6
## PC1 -0.003125858
## PC2 -0.059439783
## PC3 -0.030086322
## PC4  0.004191349
## PC5 -0.047272620
## PC6  0.238561009
## [,,9]
##              PC1          PC2           PC3           PC4           PC5
## PC1  0.314764547  0.20297323 -0.0154236206  0.0105743686  1.116786e-02
## PC2  0.202973229  0.84314134 -0.3205478486  0.0241472684 -5.404075e-02
## PC3 -0.015423621 -0.32054785  0.7649116935  0.0238224671  4.413028e-02
## PC4  0.010574369  0.02414727  0.0238224671  0.0043756124 -3.234202e-04
## PC5  0.011167858 -0.05404075  0.0441302811 -0.0003234202  9.778722e-03
## PC6  0.007211773  0.01002746  0.0008788441  0.0011084209  3.929447e-05
##              PC6
```

```
## PC1 7.211773e-03
## PC2 1.002746e-02
## PC3 8.788441e-04
## PC4 1.108421e-03
## PC5 3.929447e-05
## PC6 1.798223e-03
## [,,10]
##               PC1           PC2            PC3         PC4            PC5
## PC1  0.306719849  0.111454162  0.1045283857 -0.05598947  0.0081258744
## PC2  0.111454162  0.328056098  0.2031895750 -0.06596259 -0.0248487251
## PC3  0.104528386  0.203189575  0.1535702187 -0.04518447 -0.0151565274
## PC4 -0.055989475 -0.065962591 -0.0451844725  0.36852319  0.0274575315
## PC5  0.008125874 -0.024848725 -0.0151565274  0.02745753  0.0080753357
## PC6  0.006122415 -0.007606158  0.0005833908 -0.01116578 -0.0003370968
##                PC6
## PC1  0.0061224150
## PC2 -0.0076061578
## PC3  0.0005833908
## PC4 -0.0111657844
## PC5 -0.0003370968
## PC6  0.0048505138
##
## Hypervolume of noise component:
## 9814487
```

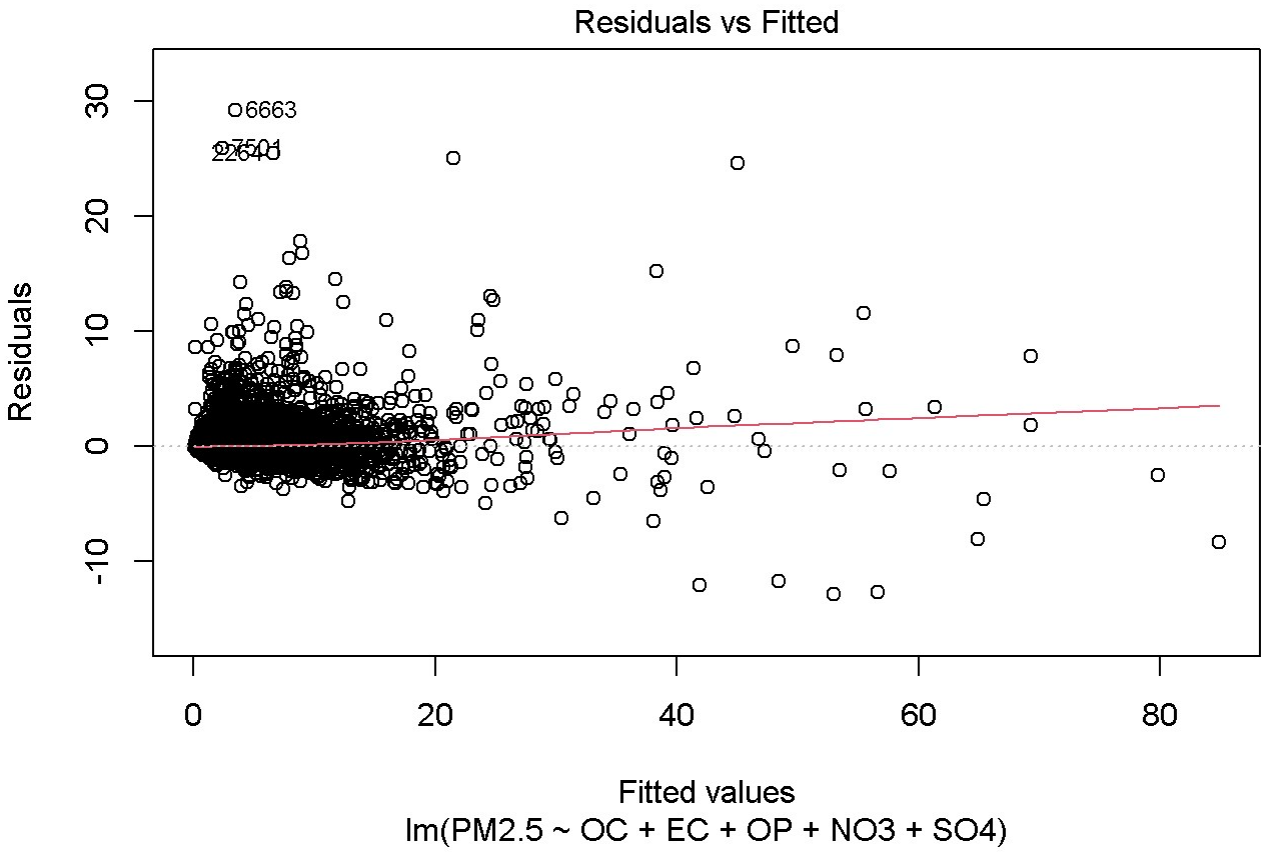
```
plot(BIC_best, what = "classification")
```

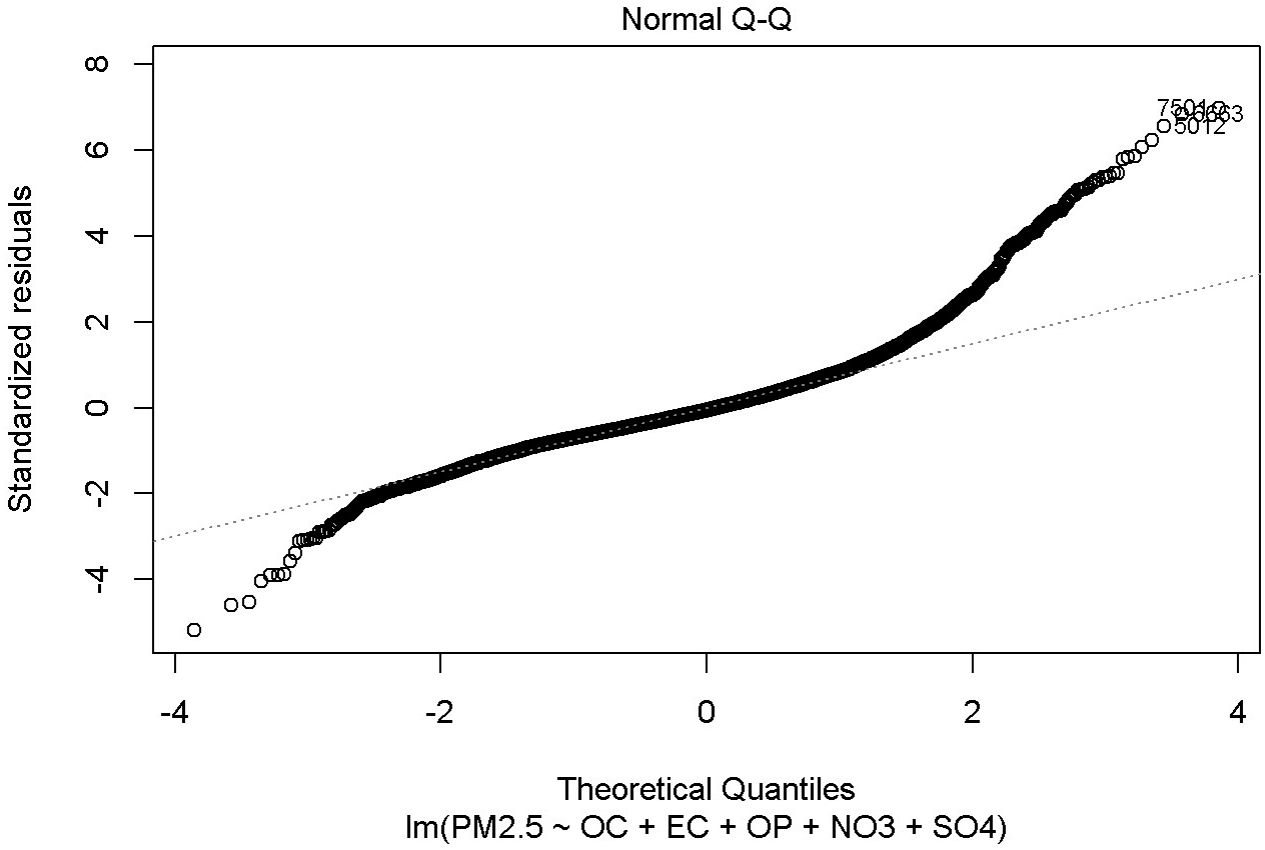


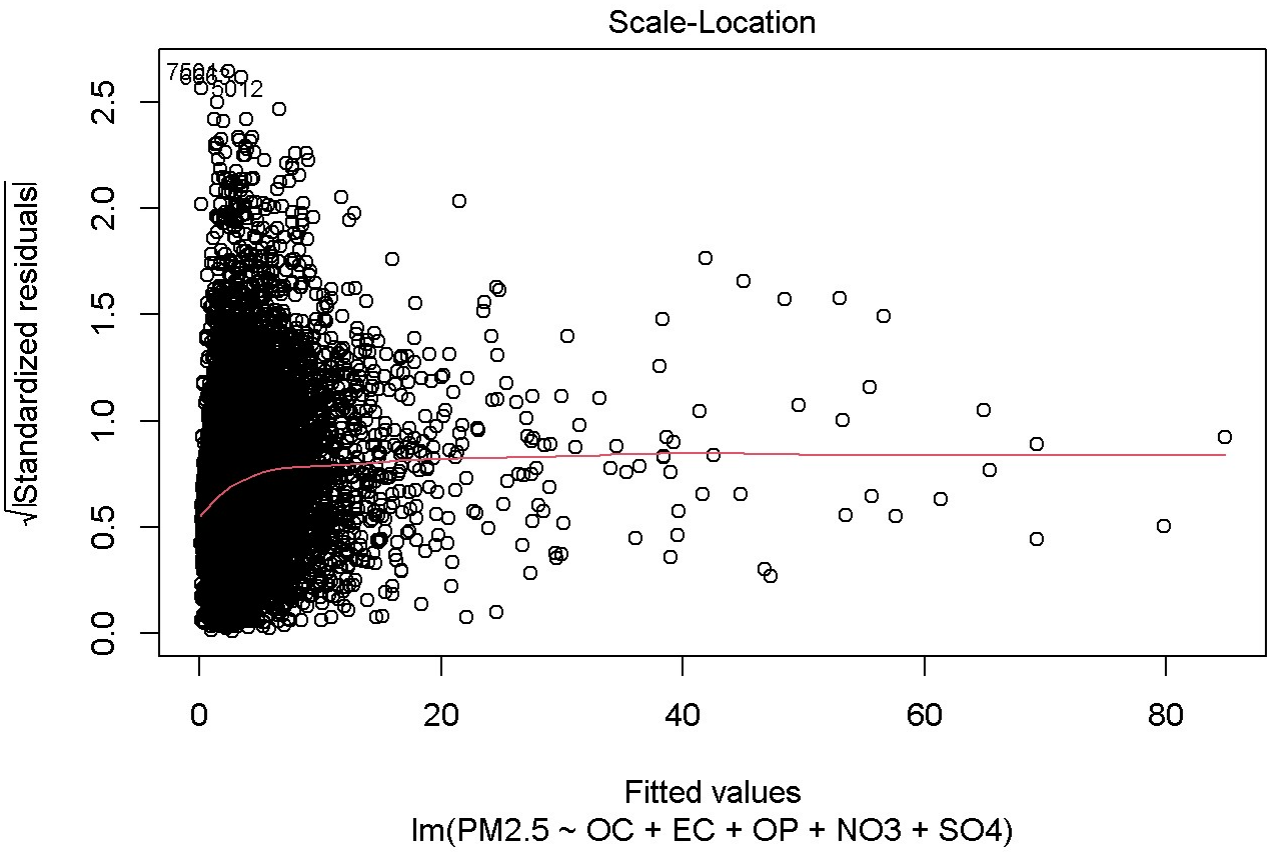

```
mod1dr<- MclustDR(BIC_best,lambda=1)
plot(mod1dr, what = "boundaries",ngrid=200)
```

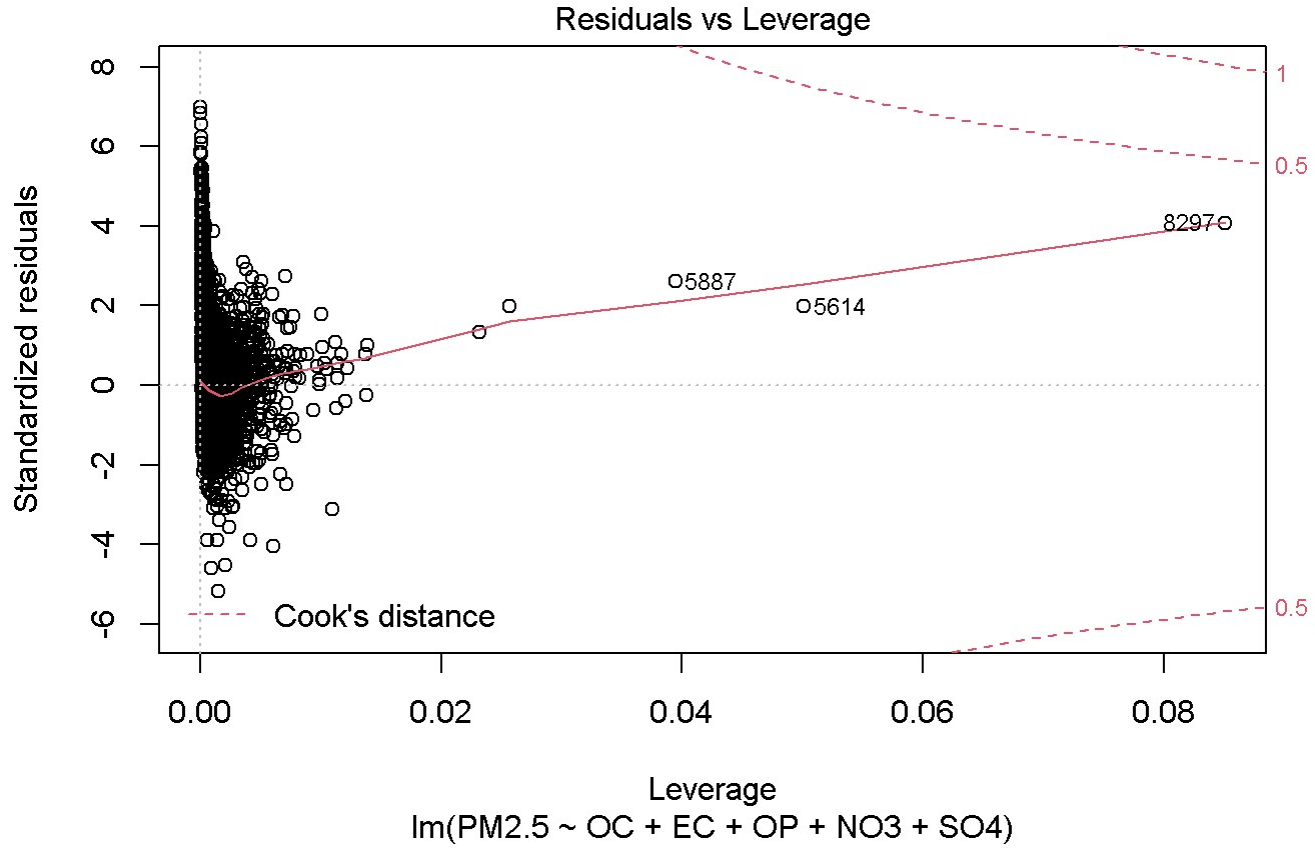



```
# --- Fit base-case weighted LS model ---

w_ii<-1/US_DATA_LRG$PM2.5_UNC^2
test_fit<-lm(PM2.5~OC+EC+OP+NO3+SO4,data=US_DATA_LRG,weights=w_ii)

plot(test_fit)
```


Residuals vs Fitted
Residuals
Fitted values
lm(PM2.5 ~ OC + EC + OP + NO3 + SO4)
6663
7501
22643


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(PM2.5 ~ OC + EC + OP + NO3 + SO4)
5012

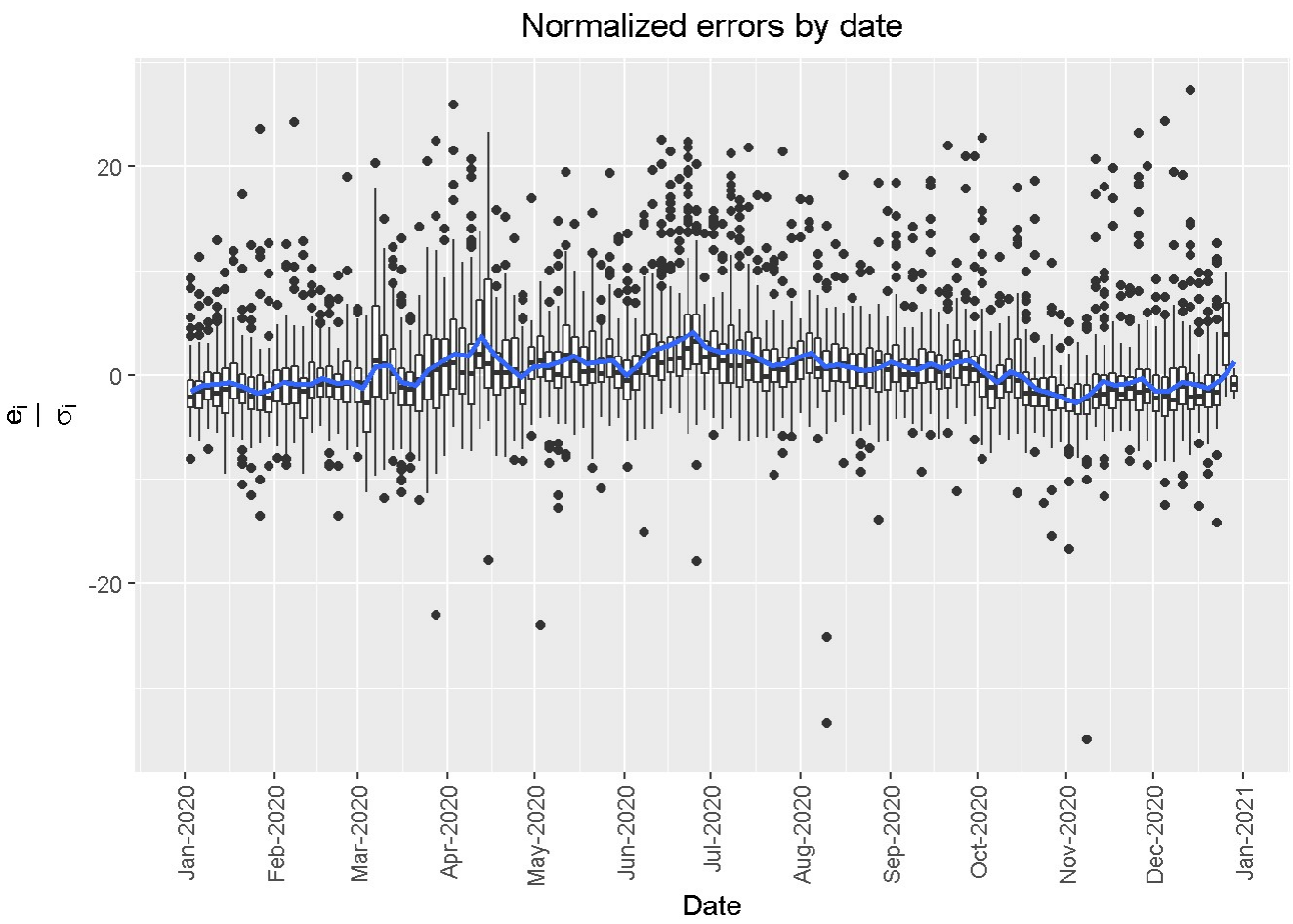

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
0
20
40
60
80
Fitted values
lm(PM2.5 ~ OC + EC + OP + NO3 + SO4)


Residuals vs Leverage
Standardized residuals
8
6
4
2
0
-2
-4
-6
1
0.5
0.5
8297
5887
5614
Cook's distance
0.00
0.02
0.04
0.06
0.08
Leverage
lm(PM2.5 ~ OC + EC + OP + NO3 + SO4)

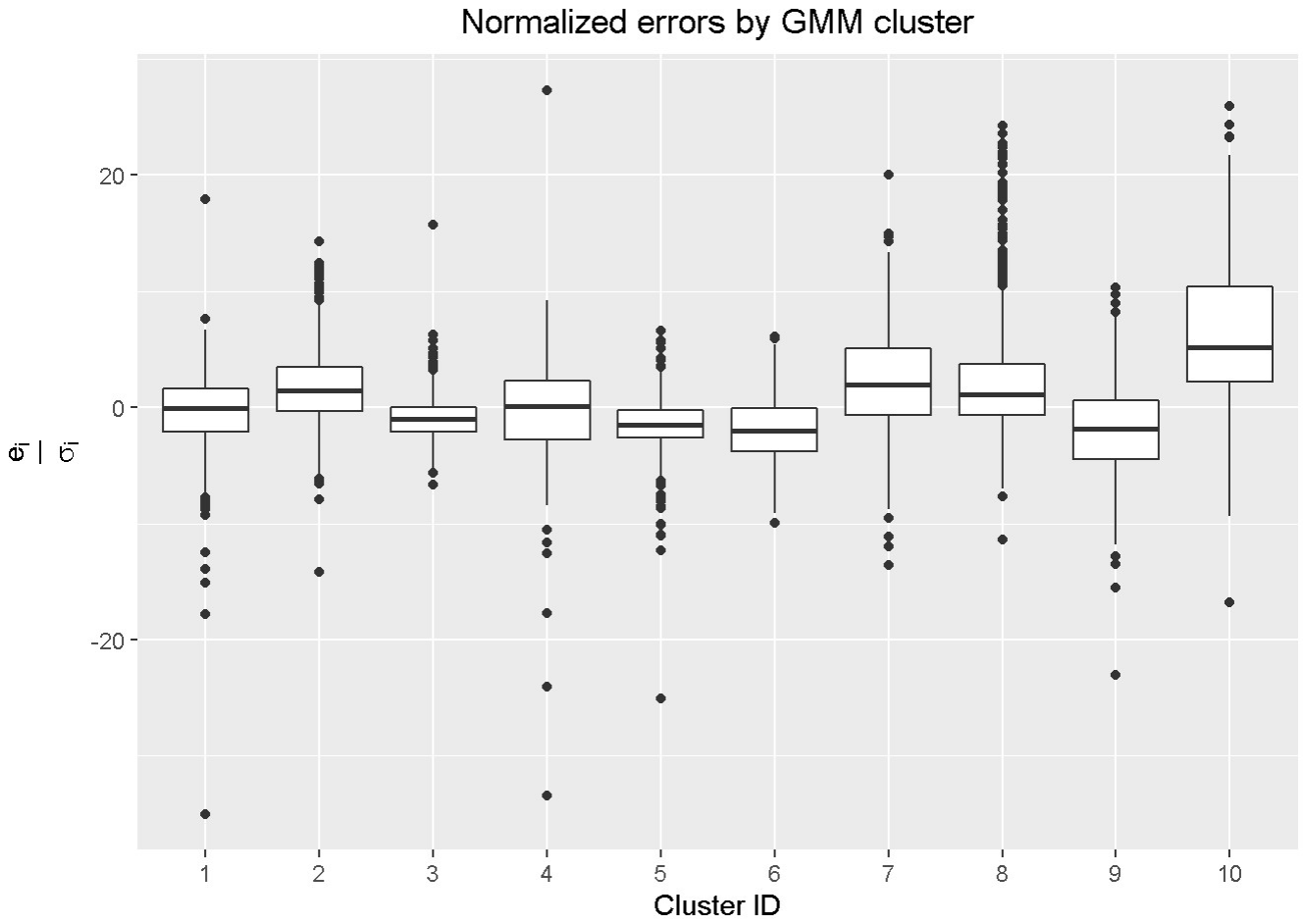
```
## --- Predict ---
PM2.5_test<-predict.lm(test_fit,US_DATA_LRG_test)

e_ii<-US_DATA_LRG_test$PM2.5-PM2.5_test
e_scaled<-e_ii/US_DATA_LRG_test$PM2.5_UNC

## --- Structure for further analysis ---
US_DATA_test_errors1<-US_DATA_LRG_test %>% dplyr::select(SiteCode,Date,PM2.5,PM2.5_
UNC) %>% mutate(Date =as.Date(Date,"%m/%d/%y"), e_test=e_ii,e_scaled=e_ii/PM2.5_UN
C)

## --- Classify test samplse with GMM ---
GMM_class_test<-predict.Mclust(BIC_best,scores_test[,1:num_PCs])

US_DATA_test_errors<-add_column(US_DATA_test_errors1,GMM_class=GMM_class_test$class
ification)
```

```
## --- Errors by date ---
ggplot(data=US_DATA_test_errors,aes(x=Date,y=e_scaled))+ geom_boxplot(aes(group=Dat
e))+
  theme(plot.title=element_text(hjust = 0.5))+geom_smooth(method= "loess",span=0.05
,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5,
hjust=1))+ylab(TeX("$\\frac{e_{i}}{\\sigma_{i}}$"))+ggtitle("Normalized errors by d
ate")+scale_x_date(breaks = "1 month", labels=scales::date_format("%b-%Y"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
## --- Errors by cluster ---
ggplot(data=US_DATA_test_errors %>% filter(GMM_class!=0),aes(x=as.factor(GMM_clas
s),y=e_scaled))+ geom_boxplot(aes(group=GMM_class))+
  theme(plot.title=element_text(hjust = 0.5))+ylab(TeX("$\\frac{e_{i}}{\\sigma_
{i}}$"))+ggtitle("Normalized errors by GMM cluster")+xlab("Cluster ID")
```

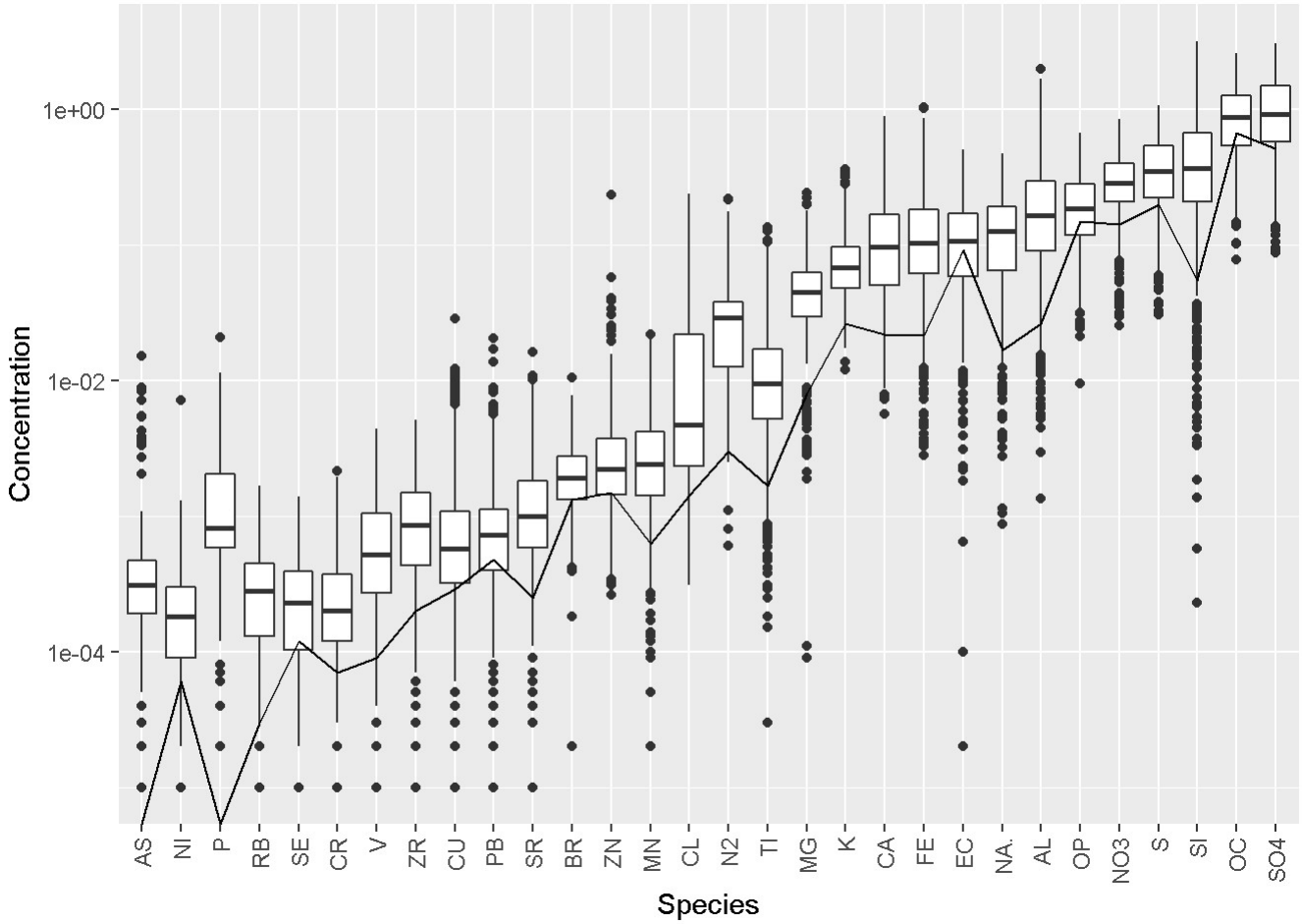



```
# ---Stack for boxplot of species on GMM cluster ---
US_DATA_test_w_stack<-stack(dplyr::select(US_DATA_LRG_test,!c("SiteCode","Date","PM
2.5_UNC","PM2.5")))

US_DATA_test_stack_GMM<-add_column(US_DATA_test_w_stack,GMM_ID =rep(GMM_class_tes
t$classification,length(levels(US_DATA_test_w_stack$ind))))

## --- Let's look only at the clusters that show high error ---
med_all_site<-apply(dplyr::select(US_DATA_LRG_test,!c("SiteCode","Date","PM2.5_UN
C","PM2.5")),2,median)
med_sites<-tibble(species=names(med_all_site),medians=med_all_site)

ggplot(US_DATA_test_stack_GMM %>% dplyr::filter(GMM_ID == c(10)), aes(x = reorder(i
nd,values,FUN=median,na.rm=TRUE), y = values)) +
  geom_boxplot()+geom_line(data=med_sites,aes(x=species,y=medians,group=1))+
  theme(plot.title=element_text(hjust = 0.5),axis.text.x = element_text(angle = 90,
vjust = 0.5, hjust=1))+ylab("Concentration")+xlab("Species")+scale_y_log10()
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 1955 rows containing non-finite values (stat_boxplot).
```




```
  #+ geom_smooth(data=US_DATA_test_stack_err_GMM,method= "loess",span=0.1 ,se=FALS
E, aes(group=1))
```

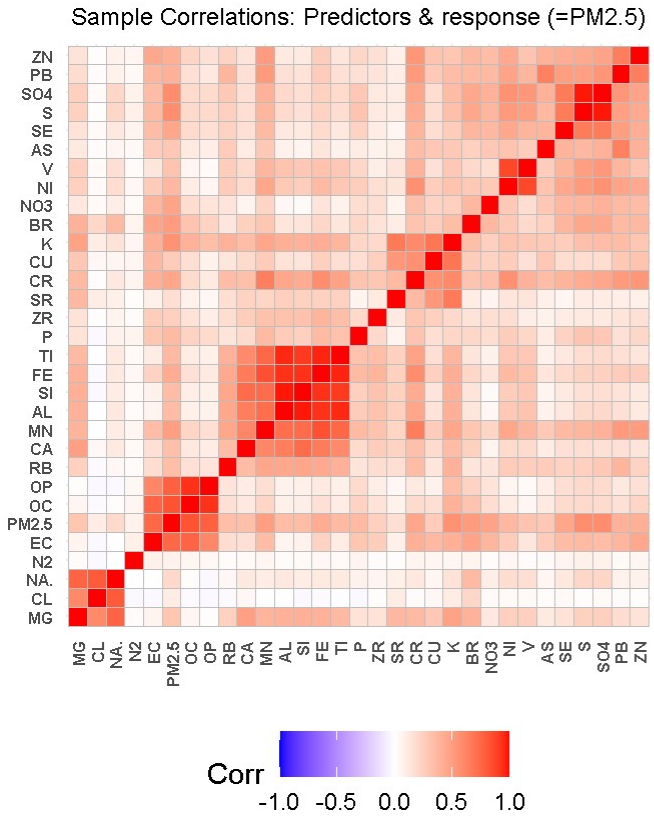
— Step 3: Figures for the Project report —

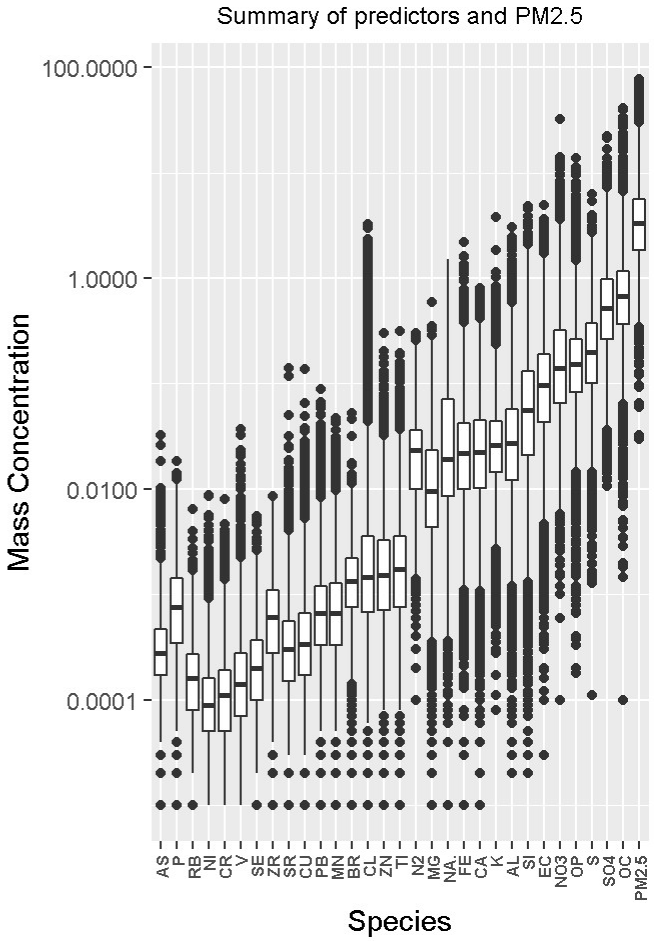
```
## --- Boxplot on normal and log scale ---
F1A<-ggcorrplot(R,hc.order=TRUE)+
  theme(legend.position = "bottom",plot.title=element_text(hjust = 0.5,size=9),axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1,size=6,face="bold"),axis.text.y = element_text(vjust = 0.5, hjust=1,size=6,face = "bold"),)+
 ggtitle("Sample Correlations: Predictors & response (=PM2.5)")

## --- Species boxplots ---
US_DATA_w_stack<-stack(dplyr::select(US_DATA_LRG,!c("SiteCode","Date","PM2.5_UNC")))

F1B<-ggplot(US_DATA_w_stack, aes(x = reorder(ind,values,FUN=median,na.rm=TRUE), y = values)) +
  geom_boxplot()+
  theme(plot.title=element_text(hjust = 0.5,size=9),axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1,size=6,face = "bold"))+ylab("Mass Concentration")+xlab("Species")+scale_y_log10(labels=scales::comma)+ggtitle("Summary of predictors and PM2.5")

grid.arrange(F1A,F1B,nrow=1)
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

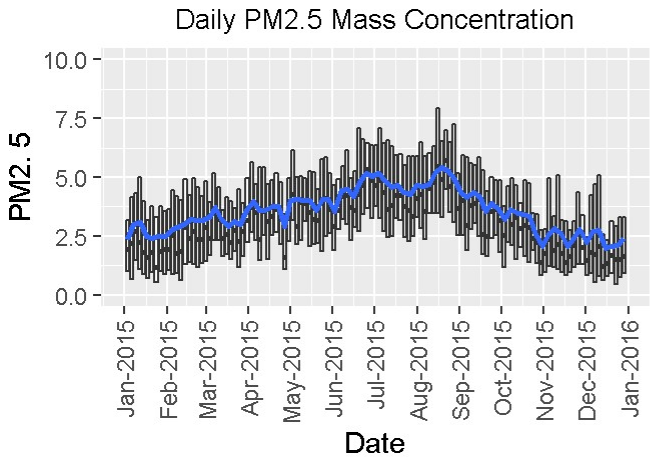
```
## Warning: Removed 37383 rows containing non-finite values (stat_boxplot).
```

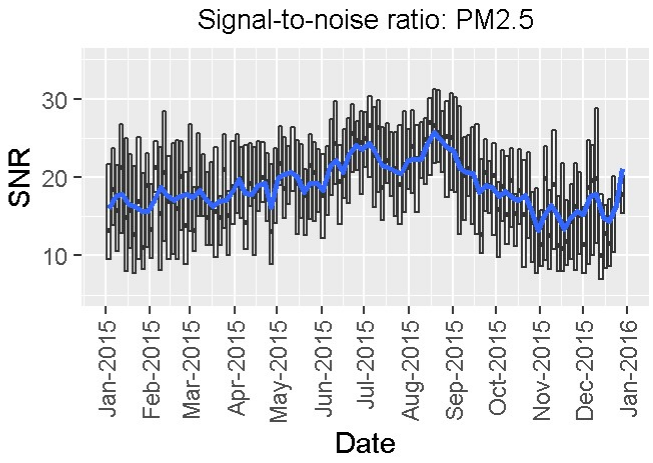

Sample Correlations: Predictors & response (=PM2.5)
ZN
PB
SO4
S
SE
AS
V
NI
NO3
BR
K
CU
CR
SR
ZR
P
TI
FE
SI
AL
MN
CA
RB
OP
OC
PM2.5
EC
N2
NA.
CL
MG
Corr
-1.0 -0.5 0.0 0.5 1.0

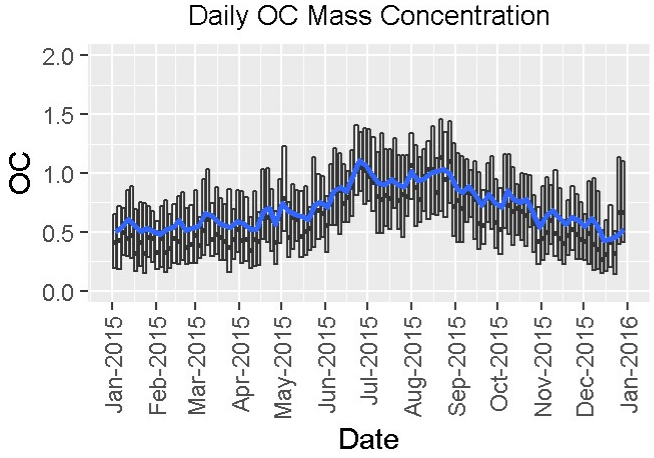

Summary of predictors and PM2.5
Mass Concentration
100.0000
1.0000
0.0100
0.0001
Species
AS P RB NI CR V SE ZR SR CU PB MN BR CL ZN TI N2 MG NA FE CA K AL SI EC NO3 OP S SO4 OC PM2.5

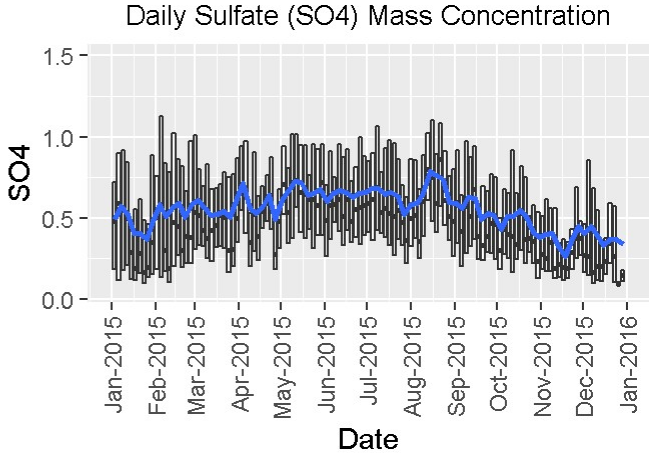
```
F2A<-ggplot(data=US_DATA_all,aes(x=as.Date(Date,"%m/%d/%Y"),y=PM2.5))+ geom_boxplot
(aes(group=as.Date(Date,"%m/%d/%Y")),outlier.shape = NA, coef = 0)+
  theme(plot.title=element_text(hjust = 0.5,size=10))+geom_smooth(method= "loess",s
pan=0.05 ,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vju
st = 0.5, hjust=1))+xlab("Date")+ylab(TeX("PM2.5"))+ggtitle("Daily PM2.5 Mass Conce
ntration")+scale_x_date(breaks = "1 month", labels=scales::date_format("%b-%Y"))+yl
im(0,10)

F2B<-ggplot(data=SNR_sort,aes(x=Date,y=SNR_PM))+ geom_boxplot(aes(group=Date),outli
er.shape = NA, coef = 0)+
  theme(plot.title=element_text(hjust = 0.5,size=10))+geom_smooth(method= "loess",s
pan=0.05 ,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vju
st = 0.5, hjust=1))+ylab(TeX("SNR"))+ggtitle("Signal-to-noise ratio: PM2.5")+scale_
x_date(breaks = "1 month", labels=scales::date_format("%b-%Y"))+ylim(5,35)

F2C<-ggplot(data=US_DATA_all,aes(x=as.Date(Date,"%m/%d/%Y"),y=OC))+ geom_boxplot(ae
s(group=as.Date(Date,"%m/%d/%Y")),outlier.shape = NA, coef = 0)+
  theme(plot.title=element_text(hjust = 0.5,size=10))+geom_smooth(method= "loess",s
pan=0.05 ,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vju
st = 0.5, hjust=1))+xlab("Date")+ylab(TeX("OC"))+ggtitle("Daily OC Mass Concentrati
on")+scale_x_date(breaks = "1 month", labels=scales::date_format("%b-%Y"))+ylim(0,
2)



F2D<-ggplot(data=US_DATA_all,aes(x=as.Date(Date,"%m/%d/%Y"),y=SO4))+ geom_boxplot(a
es(group=as.Date(Date,"%m/%d/%Y")),outlier.shape = NA, coef = 0)+
  theme(plot.title=element_text(hjust = 0.5,size=10))+geom_smooth(method= "loess",s
pan=0.05 ,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vju
st = 0.5, hjust=1))+xlab("Date")+ylab(TeX("SO4"))+ggtitle("Daily Sulfate (SO4) Mass
Concentration")+scale_x_date(breaks = "1 month", labels=scales::date_format("%b-%
Y"))+ylim(0.05,1.5)

grid.arrange(F2A,F2B,F2C,F2D,nrow=2,ncol=2)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```


Daily PM2.5 Mass Concentration
PM2.5
10.0
7.5
5.0
2.5
0.0
Jan-2015
Feb-2015
Mar-2015
Apr-2015
May-2015
Jun-2015
Jul-2015
Aug-2015
Sep-2015
Oct-2015
Nov-2015
Dec-2015
Jan-2016
Date


Signal-to-noise ratio: PM2.5
SNR
30
20
10
Jan-2015
Feb-2015
Mar-2015
Apr-2015
May-2015
Jun-2015
Jul-2015
Aug-2015
Sep-2015
Oct-2015
Nov-2015
Dec-2015
Jan-2016
Date


Daily OC Mass Concentration
OC
2.0
1.5
1.0
0.5
0.0
Jan-2015
Feb-2015
Mar-2015
Apr-2015
May-2015
Jun-2015
Jul-2015
Aug-2015
Sep-2015
Oct-2015
Nov-2015
Dec-2015
Jan-2016
Date


Daily Sulfate (SO4) Mass Concentration
SO4
1.5
1.0
0.5
0.0
Jan-2015
Feb-2015
Mar-2015
Apr-2015
May-2015
Jun-2015
Jul-2015
Aug-2015
Sep-2015
Oct-2015
Nov-2015
Dec-2015
Jan-2016
Date

Section S2: Multiple Regression Analysis Supplemental 141A Final Project

Zheyuan

12/14/2020

```
# --- Data processing and viz ---
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(broom)
library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

library(RColorBrewer)
# --- Stats---
library(corrplot)

## corrplot 0.84 loaded

library(boot)
library(mclust)

## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.

##
## Attaching package: 'mclust'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(PCAtools)
```

```
## Loading required package: ggrepel
```

```
##
## Attaching package: 'PCAtools'
```

```
## The following objects are masked from 'package:stats':
##
##     biplot, screeplot
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(Hmisc)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:boot':
##
##     melanoma
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:boot':
##
##     aml
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
# --- Spatial Analysis ---> Let's simplify our life haha
library(tmap)
library(leaflet)
#library(sp)
library(sf)
```

```
## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1
```

— Step 0: Packages to mess with –

```
    if (!requireNamespace('BiocManager', quietly = TRUE))
        install.packages('BiocManager')

    BiocManager::install('PCAtools')
```

— Step 1: Data loading and procressing —

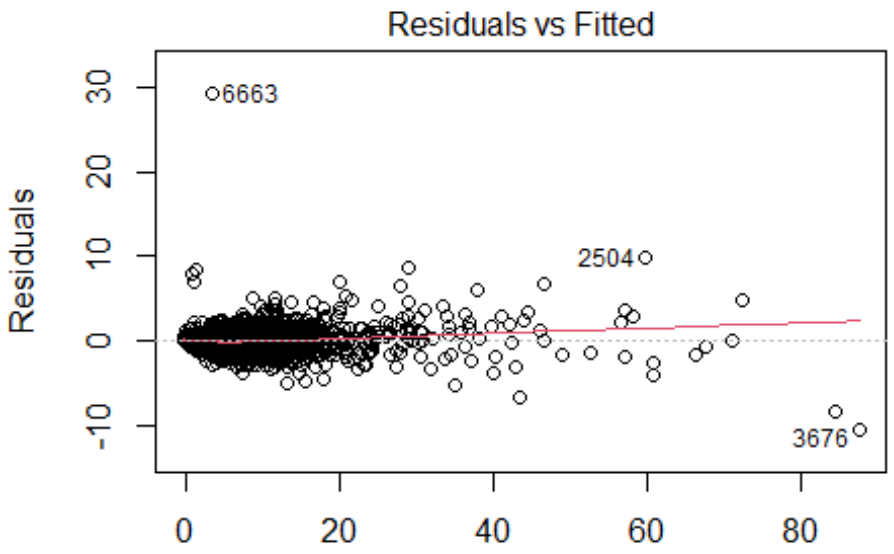
```
## --- Part a: Upload Metadata for samples ---
#path_data<-file.path(getwd(),"data")
path_data = "C:/Users/yzy/OneDrive/Documents/stats141A-FinalProject/data"
META_DATA<-as_tibble(read.csv(file.path(path_data,"IMPROVE_metadata.csv")))
## --- Filter samples from Korea and Canada ---
US_META<-META_DATA %>% filter(Country %nin% c("KR","CA"))



## --- Filter stats not in continental US ---
US_META<-META_DATA %>% filter(State %nin% c("HI","AK","VI"))

## -- Use Mississippi River as a dividing point for WEst-East US --
MR_coords<-c(47.239722, -95.2075)
POS_Sampler<-as.numeric(US_META$Longitude <MR_coords[2])
# --- 1 are WEst US, 0 are East
US_META<-add_column(US_META,WE_US = POS_Sampler)

## --- Part b: Load samples data ---
```

```
DATA<-
as_tibble(read.csv(file.path(path_data,"IMPROVE_2015_data_w_UNC_v2.csv")))

## --- Part c: Select samples from SW given site identifiers from SW_META
table ("Code")
US_DATA_all<-as_tibble(DATA %>% filter(SiteCode %in% US_META$Code))

# Let's identify any samples that (grossly) violate PM2.5 mass balances
# PM2.5 (=Y) cannot be negative!
# Since there's some probability that PM2.5 is negative due to errors at low
concentration, we may use PM2.5 uncertainties to remove samples that fall
outside -3*PM2.5_UNC.
# In this way, we don't risk censoring the data but do remove likely
erroneous data.
US_DATA_all<-US_DATA_all %>% dplyr::filter(PM2.5 > -3*PM2.5_UNC)

exclude<-
c("PM10","POC","ammNO3","ammSO4","SOIL","SeaSalt","OC1","OC2","OC3","OC4","EC
1","EC2","EC3","fAbs_MDL","fAbs")
US_DATA_LRG<- US_DATA_all %>% dplyr::select(!contains(exclude) &
!matches("_UNC") | matches("PM2.5_UNC"))
any(is.na(US_DATA_LRG))

## [1] TRUE

US_DATA_LRG<-US_DATA_LRG[which(complete.cases(US_DATA_LRG)),]
any(is.na(US_DATA_LRG))

## [1] FALSE

## --- Instead of random partitioning, I will partition by first sorting
samples by SiteCode and DATE (already done) and place every other sample in
the test set.
# --- This data has seasonality. Sorting by date therefore ensures
seasonality is equivalent between datasets
n<-nrow(US_DATA_LRG)
ind_test<-seq(1,n,2)
US_DATA_LRG_test<-US_DATA_LRG[ind_test,]
US_DATA_LRG<-US_DATA_LRG[-ind_test,]
```

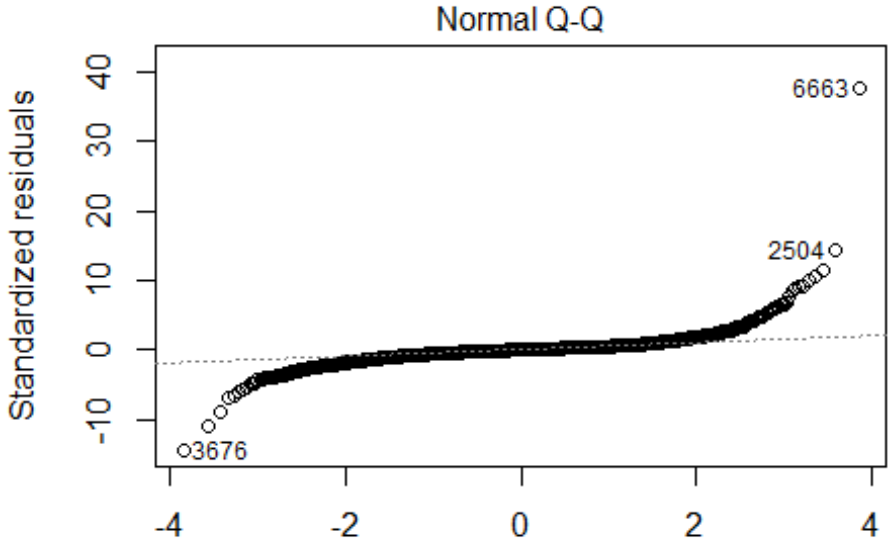
#Rgression Analysis

```
#First order model
fit = lm(PM2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU + FE + PB +
MG + MN + NI + N2 + P + K + RB + SE + SI + NA. + SR + S + TI + V + ZN + ZR +
NO3 + SO4, data = US_DATA_LRG)
summary(fit)

##
## Call:
## lm(formula = PM2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL +
##     CR + CU + FE + PB + MG + MN + NI + N2 + P + K + RB + SE +
```

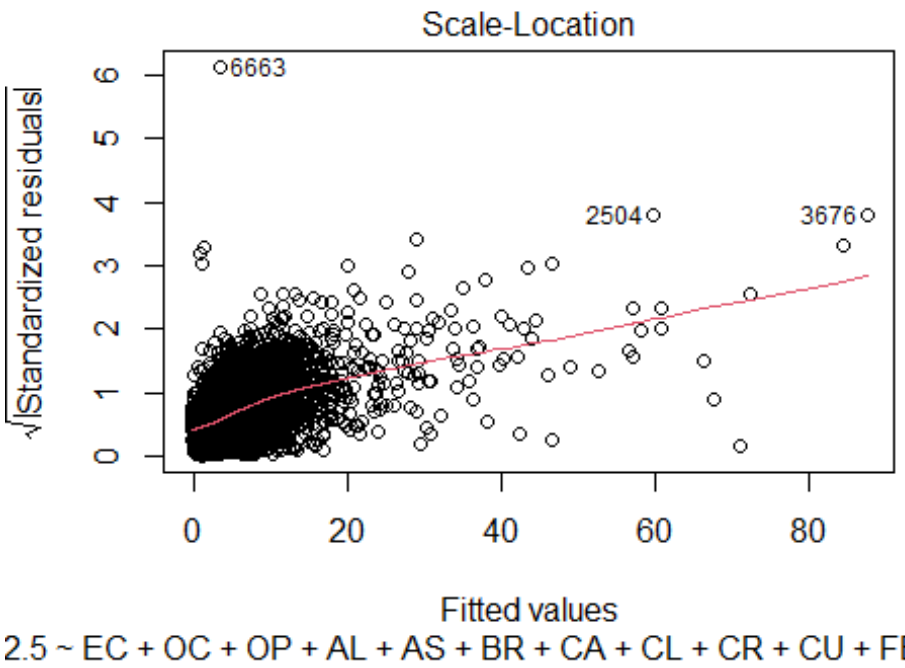
```
##     SI + NA. + SR + S + TI + V + ZN + ZR + NO3 + SO4, data = US_DATA_LRG)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3934  -0.2615   0.0169   0.2508  29.1786
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.21256    0.01491 -14.256  < 2e-16 ***
## EC            -0.11101    0.08255  -1.345  0.17874
## OC             1.93476    0.01933 100.074  < 2e-16 ***
## OP             0.22448    0.05640   3.980 6.94e-05 ***
## AL            -0.58137    0.49521  -1.174  0.24043
## AS            16.62336   16.38978   1.014  0.31049
## BR             5.48740    6.90699   0.794  0.42694
## CA             1.91323    0.25723   7.438 1.12e-13 ***
## CL             3.45763    0.10375  33.325  < 2e-16 ***
## CR          -148.03756   58.86488  -2.515  0.01193 *
## CU           -26.39870    5.83226  -4.526 6.08e-06 ***
## FE             3.65516    0.75521   4.840 1.32e-06 ***
## PB            24.95061    5.76302   4.329 1.51e-05 ***
## MG            -0.03643    0.76443  -0.048  0.96199
## MN           -20.57501   10.10126  -2.037  0.04169 *
## NI            49.78920   78.64428   0.633  0.52669
## N2             0.04225    0.33241   0.127  0.89886
## P             44.97508    9.19560   4.891 1.02e-06 ***
## K              2.96531    0.29044  10.210  < 2e-16 ***
## RB            62.93135   42.41997   1.484  0.13797
## SE           144.02218   36.45795   3.950 7.87e-05 ***
## SI             2.99262    0.26512  11.288  < 2e-16 ***
## NA.            0.11404    0.17508   0.651  0.51484
## SR           -14.43224    5.46132  -2.643  0.00824 **
## S              3.92478    0.16637  23.591  < 2e-16 ***
## TI            17.30420    5.30792   3.260  0.00112 **
## V             25.86340   21.44322   1.206  0.22780
## ZN            -0.55866    1.58575  -0.352  0.72462
## ZR             9.27636   10.84616   0.855  0.39243
## NO3            1.22060    0.01208 101.025  < 2e-16 ***
## SO4            0.39605    0.05671   6.984 3.09e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7764 on 8616 degrees of freedom
## Multiple R-squared:  0.9763, Adjusted R-squared:  0.9762
## F-statistic: 1.182e+04 on 30 and 8616 DF,  p-value: < 2.2e-16

#Assumption check
plot(fit)
```



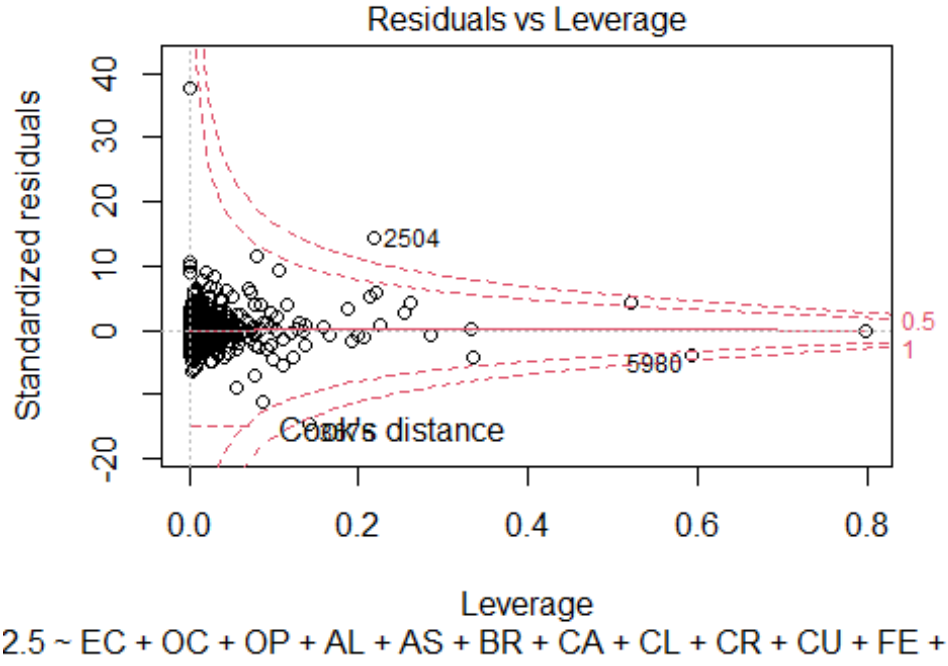
2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU + FE + P



2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU + FE + P



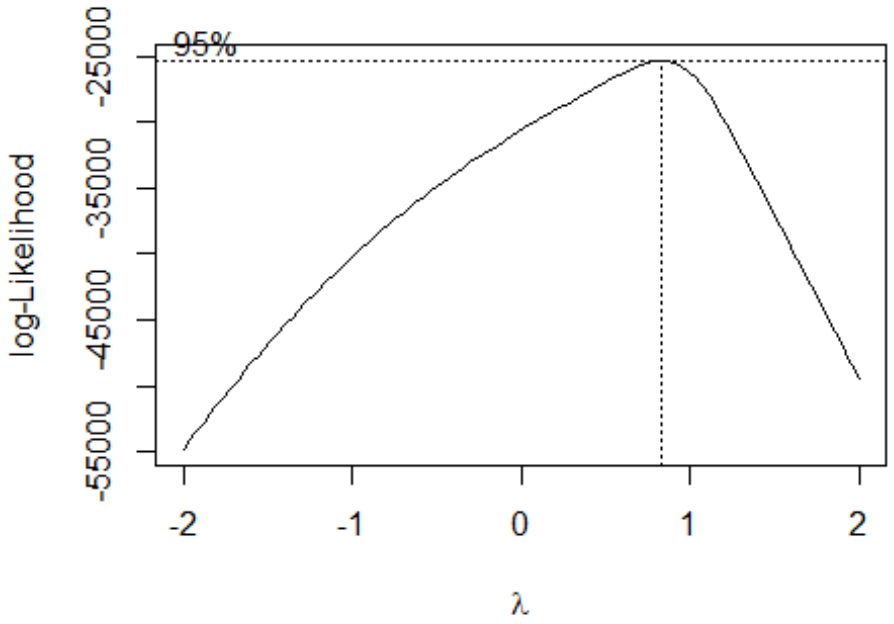
2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU + FE + P



2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU + FE + P

```
#Box Cox Procedure
min(US_DATA_LRG$PM2.5)

## [1] -0.093
```

```
fit.b = lm(PM2.5 + 0.26 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU +
FE + PB + MG + MN + NI + N2 + P + K + RB + SE + SI + NA. + SR + S + TI + V +
ZN + ZR + NO3 + SO4, data = US_DATA_LRG)
boxcox(fit.b)
```



#The QQ plot looks strange, but that just because there are several outliers. The lambda value in Box Cox procedure is very close to 1, which means we do not need to transform PM2.5 to make it more normal. The assumption of homoscedasticity and nonlinearity are valid, too.

```
#model selection
fit0 = lm(PM2.5 ~ 1, data = US_DATA_LRG)
#forward selection on AIC
mod1 = stepAIC(fit0, scope = list(upper = fit, lower = fit0), direction =
"forward", k = 2, trace = FALSE)
#backward elimination on AIC
mod2 = stepAIC(fit, scope = list(upper = fit, lower = fit0), direction =
"backward", k = 2, trace = FALSE)
#forward stepwise on AIC
mod3 = stepAIC(fit0, scope = list(upper = fit, lower = fit0), direction =
"both", k = 2, trace = FALSE)
#backward stepwise on AIC
mod4 = stepAIC(fit, scope = list(upper = fit, lower = fit0), direction =
"forward", k = 2, trace = FALSE)
#forward selection on BIC
mod5 = stepAIC(fit0, scope = list(upper = fit, lower = fit0), direction =
"forward", k = log(n), trace = FALSE)
#backward elimination on BIC
```

```r
mod6 = stepAIC(fit, scope = list(upper = fit, lower = fit0), direction =
"backward", k = log(n), trace = FALSE)
#forward stepwise on BIC
mod7 = stepAIC(fit0, scope = list(upper = fit, lower = fit0), direction =
"both", k = log(n), trace = FALSE)
#backward stepwise on BIC
mod8 = stepAIC(fit, scope = list(upper = fit, lower = fit0), direction =
"forward", k = log(n), trace = FALSE)
summary(mod1)

##
## Call:
## lm(formula = PM2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K +
##     CA + CU + PB + P + OP + TI + SE + V + CR + SR + MN + RB,
##     data = US_DATA_LRG)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3521  -0.2623   0.0182   0.2524  29.1815
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.20458    0.01387 -14.754  < 2e-16 ***
## OC             1.92315    0.01473 130.522  < 2e-16 ***
## SO4            0.39926    0.05640   7.079 1.56e-12 ***
## FE             3.71932    0.69901   5.321 1.06e-07 ***
## NO3            1.22098    0.01175 103.883  < 2e-16 ***
## CL             3.53683    0.05907  59.880  < 2e-16 ***
## SI             2.76513    0.13729  20.141  < 2e-16 ***
## S              3.91824    0.16341  23.978  < 2e-16 ***
## K              2.95832    0.27518  10.750  < 2e-16 ***
## CA             2.02114    0.22873   8.836  < 2e-16 ***
## CU           -26.07097    5.64571  -4.618 3.93e-06 ***
## PB            26.11871    4.86269   5.371 8.02e-08 ***
## P             45.10306    9.17202   4.917 8.93e-07 ***
## OP             0.23829    0.05158   4.619 3.90e-06 ***
## TI            15.02221    4.34099   3.461 0.000542 ***
## SE           146.77176   36.11467   4.064 4.87e-05 ***
## V             38.32761   11.37525   3.369 0.000757 ***
## CR          -154.91486   53.38461  -2.902 0.003719 **
## SR           -15.10595    5.36982  -2.813 0.004917 **
## MN           -22.34431    9.61670  -2.323 0.020176 *
## RB            63.64680   42.02435   1.515 0.129930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7763 on 8626 degrees of freedom
## Multiple R-squared:  0.9763, Adjusted R-squared:  0.9762
## F-statistic: 1.774e+04 on 20 and 8626 DF,  p-value: < 2.2e-16
```

```r
summary(mod2)
```

```
##
## Call:
## lm(formula = PM2.5 ~ OC + OP + CA + CL + CR + CU + FE + PB +
##     MN + P + K + RB + SE + SI + SR + S + TI + V + NO3 + SO4,
##     data = US_DATA_LRG)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3521  -0.2623   0.0182   0.2524  29.1815
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.20458    0.01387 -14.754  < 2e-16 ***
## OC             1.92315    0.01473 130.522  < 2e-16 ***
## OP             0.23829    0.05158   4.619 3.90e-06 ***
## CA             2.02114    0.22873   8.836  < 2e-16 ***
## CL             3.53683    0.05907  59.880  < 2e-16 ***
## CR          -154.91486   53.38461  -2.902 0.003719 **
## CU           -26.07097    5.64571  -4.618 3.93e-06 ***
## FE             3.71932    0.69901   5.321 1.06e-07 ***
## PB            26.11871    4.86269   5.371 8.02e-08 ***
## MN           -22.34431    9.61670  -2.323 0.020176 *
## P             45.10306    9.17202   4.917 8.93e-07 ***
## K              2.95832    0.27518  10.750  < 2e-16 ***
## RB            63.64680   42.02435   1.515 0.129930
## SE           146.77176   36.11467   4.064 4.87e-05 ***
## SI             2.76513    0.13729  20.141  < 2e-16 ***
## SR           -15.10595    5.36982  -2.813 0.004917 **
## S              3.91824    0.16341  23.978  < 2e-16 ***
## TI            15.02221    4.34099   3.461 0.000542 ***
## V             38.32761   11.37525   3.369 0.000757 ***
## NO3            1.22098    0.01175 103.883  < 2e-16 ***
## SO4            0.39926    0.05640   7.079 1.56e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7763 on 8626 degrees of freedom
## Multiple R-squared:  0.9763, Adjusted R-squared:  0.9762
## F-statistic: 1.774e+04 on 20 and 8626 DF,  p-value: < 2.2e-16
```

```r
summary(mod3)
```

```
##
## Call:
## lm(formula = PM2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K +
##     CA + CU + PB + P + OP + TI + SE + V + CR + SR + MN + RB,
##     data = US_DATA_LRG)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3521  -0.2623   0.0182   0.2524  29.1815
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.20458    0.01387 -14.754  < 2e-16 ***
## OC             1.92315    0.01473 130.522  < 2e-16 ***
## SO4            0.39926    0.05640   7.079 1.56e-12 ***
## FE             3.71932    0.69901   5.321 1.06e-07 ***
## NO3            1.22098    0.01175 103.883  < 2e-16 ***
## CL             3.53683    0.05907  59.880  < 2e-16 ***
## SI             2.76513    0.13729  20.141  < 2e-16 ***
## S              3.91824    0.16341  23.978  < 2e-16 ***
## K              2.95832    0.27518  10.750  < 2e-16 ***
## CA             2.02114    0.22873   8.836  < 2e-16 ***
## CU           -26.07097    5.64571  -4.618 3.93e-06 ***
## PB            26.11871    4.86269   5.371 8.02e-08 ***
## P             45.10306    9.17202   4.917 8.93e-07 ***
## OP             0.23829    0.05158   4.619 3.90e-06 ***
## TI            15.02221    4.34099   3.461 0.000542 ***
## SE           146.77176   36.11467   4.064 4.87e-05 ***
## V             38.32761   11.37525   3.369 0.000757 ***
## CR          -154.91486   53.38461  -2.902 0.003719 **
## SR           -15.10595    5.36982  -2.813 0.004917 **
## MN           -22.34431    9.61670  -2.323 0.020176 *
## RB            63.64680   42.02435   1.515 0.129930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7763 on 8626 degrees of freedom
## Multiple R-squared:  0.9763, Adjusted R-squared:  0.9762
## F-statistic: 1.774e+04 on 20 and 8626 DF,  p-value: < 2.2e-16
```

```
summary(mod4)
```

```
##
## Call:
## lm(formula = PM2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL +
##     CR + CU + FE + PB + MG + MN + NI + N2 + P + K + RB + SE +
##     SI + NA. + SR + S + TI + V + ZN + ZR + NO3 + SO4, data = US_DATA_LRG)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3934  -0.2615   0.0169   0.2508  29.1786
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.21256    0.01491 -14.256  < 2e-16 ***
## EC            -0.11101    0.08255  -1.345  0.17874
```

```
## OC             1.93476    0.01933 100.074  < 2e-16 ***
## OP             0.22448    0.05640   3.980 6.94e-05 ***
## AL            -0.58137    0.49521  -1.174  0.24043
## AS            16.62336   16.38978   1.014  0.31049
## BR             5.48740    6.90699   0.794  0.42694
## CA             1.91323    0.25723   7.438 1.12e-13 ***
## CL             3.45763    0.10375  33.325  < 2e-16 ***
## CR          -148.03756   58.86488  -2.515  0.01193 *
## CU           -26.39870    5.83226  -4.526 6.08e-06 ***
## FE             3.65516    0.75521   4.840 1.32e-06 ***
## PB            24.95061    5.76302   4.329 1.51e-05 ***
## MG            -0.03643    0.76443  -0.048  0.96199
## MN           -20.57501   10.10126  -2.037  0.04169 *
## NI            49.78920   78.64428   0.633  0.52669
## N2             0.04225    0.33241   0.127  0.89886
## P             44.97508    9.19560   4.891 1.02e-06 ***
## K              2.96531    0.29044  10.210  < 2e-16 ***
## RB            62.93135   42.41997   1.484  0.13797
## SE           144.02218   36.45795   3.950 7.87e-05 ***
## SI             2.99262    0.26512  11.288  < 2e-16 ***
## NA.            0.11404    0.17508   0.651  0.51484
## SR           -14.43224    5.46132  -2.643  0.00824 **
## S              3.92478    0.16637  23.591  < 2e-16 ***
## TI            17.30420    5.30792   3.260  0.00112 **
## V             25.86340   21.44322   1.206  0.22780
## ZN            -0.55866    1.58575  -0.352  0.72462
## ZR             9.27636   10.84616   0.855  0.39243
## NO3            1.22060    0.01208 101.025  < 2e-16 ***
## SO4            0.39605    0.05671   6.984 3.09e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7764 on 8616 degrees of freedom
## Multiple R-squared:  0.9763, Adjusted R-squared:  0.9762
## F-statistic: 1.182e+04 on 30 and 8616 DF,  p-value: < 2.2e-16
```

```
summary(mod5)
```

```
##
## Call:
## lm(formula = PM2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K +
##     CA + CU + PB + P + OP + TI + SE, data = US_DATA_LRG)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2069  -0.2578   0.0215   0.2512  29.2244
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.20680    0.01373 -15.065  < 2e-16 ***
```

```
## OC            1.93015    0.01466 131.678  < 2e-16 ***
## SO4           0.42689    0.05485   7.783 7.92e-15 ***
## FE            2.26146    0.58252   3.882 0.000104 ***
## NO3           1.22387    0.01164 105.154  < 2e-16 ***
## CL            3.53975    0.05901  59.987  < 2e-16 ***
## SI            2.99026    0.13068  22.882  < 2e-16 ***
## S             3.86775    0.16145  23.957  < 2e-16 ***
## K             2.43625    0.22489  10.833  < 2e-16 ***
## CA            1.91137    0.22327   8.561  < 2e-16 ***
## CU          -29.52848    5.33513  -5.535 3.21e-08 ***
## PB           24.50111    4.41828   5.545 3.02e-08 ***
## P            47.15010    9.16384   5.145 2.73e-07 ***
## OP            0.22895    0.05157   4.439 9.14e-06 ***
## TI           18.45628    4.19983   4.395 1.12e-05 ***
## SE          141.35127   35.87357   3.940 8.20e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7777 on 8631 degrees of freedom
## Multiple R-squared:  0.9762, Adjusted R-squared:  0.9761
## F-statistic: 2.357e+04 on 15 and 8631 DF,  p-value: < 2.2e-16
```

```
summary(mod6)
```

```
##
## Call:
## lm(formula = PM2.5 ~ OC + OP + CA + CL + CR + CU + FE + PB +
##     P + K + SE + SI + S + TI + V + NO3 + SO4, data = US_DATA_LRG)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2598  -0.2620   0.0178   0.2521  29.1899
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.20364    0.01382 -14.737  < 2e-16 ***
## OC             1.92904    0.01465 131.712  < 2e-16 ***
## OP             0.23141    0.05158   4.486 7.34e-06 ***
## CA             1.87043    0.22448   8.332  < 2e-16 ***
## CL             3.53383    0.05896  59.932  < 2e-16 ***
## CR          -170.07023   53.03200  -3.207 0.001346 **
## CU           -25.86852    5.46109  -4.737 2.20e-06 ***
## FE             3.00058    0.61958   4.843 1.30e-06 ***
## PB            25.30822    4.47082   5.661 1.56e-08 ***
## P             45.47777    9.16990   4.959 7.20e-07 ***
## K              2.56550    0.22996  11.156  < 2e-16 ***
## SE           141.90823   35.99421   3.943 8.13e-05 ***
## SI             2.89006    0.13273  21.774  < 2e-16 ***
## S              3.89311    0.16287  23.903  < 2e-16 ***
## TI            15.80104    4.25261   3.716 0.000204 ***
```

```
## V             37.01644   11.37785   3.253 0.001145 **
## NO3            1.22619    0.01165 105.220  < 2e-16 ***
## SO4            0.40972    0.05620   7.290 3.38e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7769 on 8629 degrees of freedom
## Multiple R-squared:  0.9762, Adjusted R-squared:  0.9762
## F-statistic: 2.084e+04 on 17 and 8629 DF,  p-value: < 2.2e-16
```

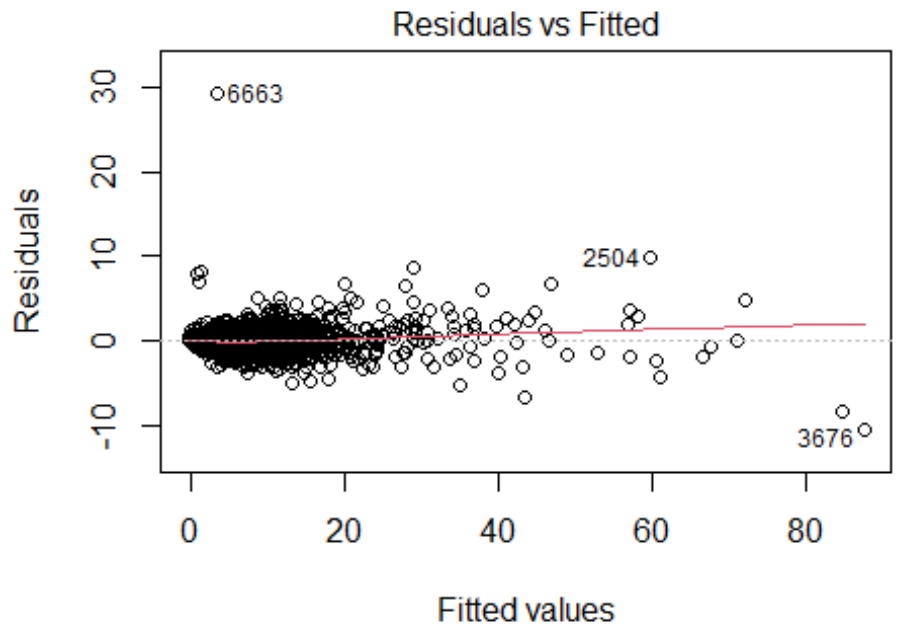
```
summary(mod7)
```

```
##
## Call:
## lm(formula = PM2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K +
##     CA + CU + PB + P + OP + TI + SE, data = US_DATA_LRG)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2069  -0.2578   0.0215   0.2512  29.2244
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.20680    0.01373 -15.065  < 2e-16 ***
## OC            1.93015    0.01466 131.678  < 2e-16 ***
## SO4           0.42689    0.05485   7.783 7.92e-15 ***
## FE            2.26146    0.58252   3.882 0.000104 ***
## NO3           1.22387    0.01164 105.154  < 2e-16 ***
## CL            3.53975    0.05901  59.987  < 2e-16 ***
## SI            2.99026    0.13068  22.882  < 2e-16 ***
## S             3.86775    0.16145  23.957  < 2e-16 ***
## K             2.43625    0.22489  10.833  < 2e-16 ***
## CA            1.91137    0.22327   8.561  < 2e-16 ***
## CU          -29.52848    5.33513  -5.535 3.21e-08 ***
## PB           24.50111    4.41828   5.545 3.02e-08 ***
## P            47.15010    9.16384   5.145 2.73e-07 ***
## OP            0.22895    0.05157   4.439 9.14e-06 ***
## TI           18.45628    4.19983   4.395 1.12e-05 ***
## SE          141.35127   35.87357   3.940 8.20e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7777 on 8631 degrees of freedom
## Multiple R-squared:  0.9762, Adjusted R-squared:  0.9761
## F-statistic: 2.357e+04 on 15 and 8631 DF,  p-value: < 2.2e-16
```

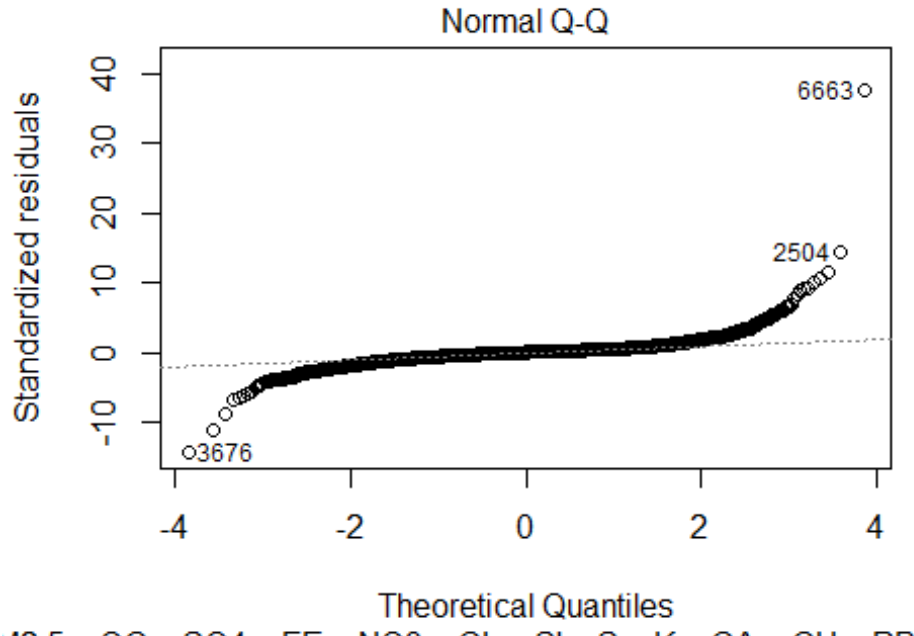
```
summary(mod8)
```

```
##
## Call:
## lm(formula = PM2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL +
```

```
##     CR + CU + FE + PB + MG + MN + NI + N2 + P + K + RB + SE +
##     SI + NA. + SR + S + TI + V + ZN + ZR + NO3 + SO4, data = US_DATA_LRG)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3934  -0.2615   0.0169   0.2508  29.1786
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.21256    0.01491 -14.256  < 2e-16 ***
## EC            -0.11101    0.08255  -1.345  0.17874
## OC             1.93476    0.01933 100.074  < 2e-16 ***
## OP             0.22448    0.05640   3.980 6.94e-05 ***
## AL            -0.58137    0.49521  -1.174  0.24043
## AS            16.62336   16.38978   1.014  0.31049
## BR             5.48740    6.90699   0.794  0.42694
## CA             1.91323    0.25723   7.438 1.12e-13 ***
## CL             3.45763    0.10375  33.325  < 2e-16 ***
## CR          -148.03756   58.86488  -2.515  0.01193 *
## CU           -26.39870    5.83226  -4.526 6.08e-06 ***
## FE             3.65516    0.75521   4.840 1.32e-06 ***
## PB            24.95061    5.76302   4.329 1.51e-05 ***
## MG            -0.03643    0.76443  -0.048  0.96199
## MN           -20.57501   10.10126  -2.037  0.04169 *
## NI            49.78920   78.64428   0.633  0.52669
## N2             0.04225    0.33241   0.127  0.89886
## P             44.97508    9.19560   4.891 1.02e-06 ***
## K              2.96531    0.29044  10.210  < 2e-16 ***
## RB            62.93135   42.41997   1.484  0.13797
## SE           144.02218   36.45795   3.950 7.87e-05 ***
## SI             2.99262    0.26512  11.288  < 2e-16 ***
## NA.            0.11404    0.17508   0.651  0.51484
## SR           -14.43224    5.46132  -2.643  0.00824 **
## S              3.92478    0.16637  23.591  < 2e-16 ***
## TI            17.30420    5.30792   3.260  0.00112 **
## V             25.86340   21.44322   1.206  0.22780
## ZN            -0.55866    1.58575  -0.352  0.72462
## ZR             9.27636   10.84616   0.855  0.39243
## NO3            1.22060    0.01208 101.025  < 2e-16 ***
## SO4            0.39605    0.05671   6.984 3.09e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7764 on 8616 degrees of freedom
## Multiple R-squared:  0.9763, Adjusted R-squared:  0.9762
## F-statistic: 1.182e+04 on 30 and 8616 DF,  p-value: < 2.2e-16

plot(mod1, which = c(1,2))
```

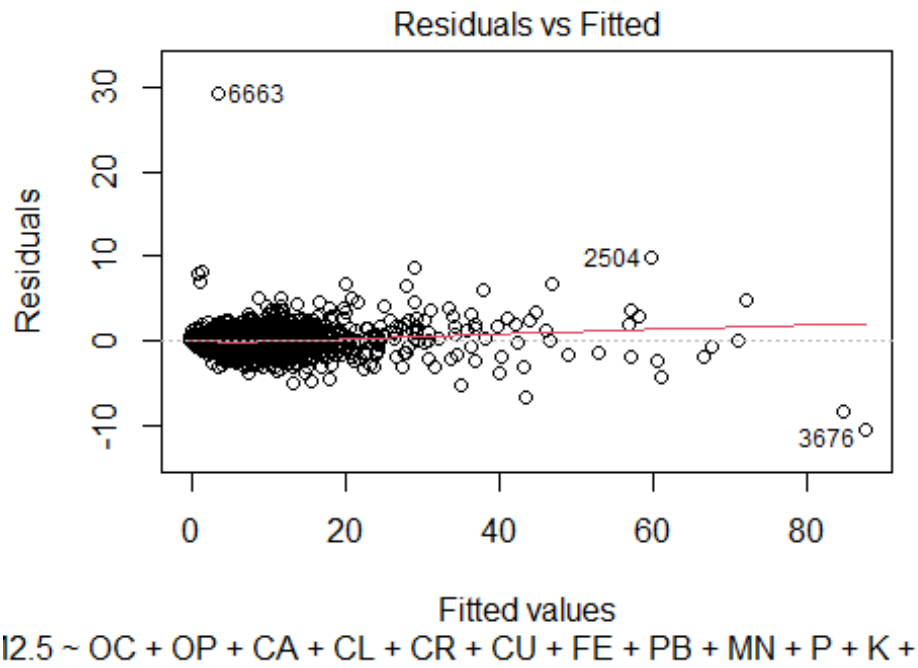


M2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P

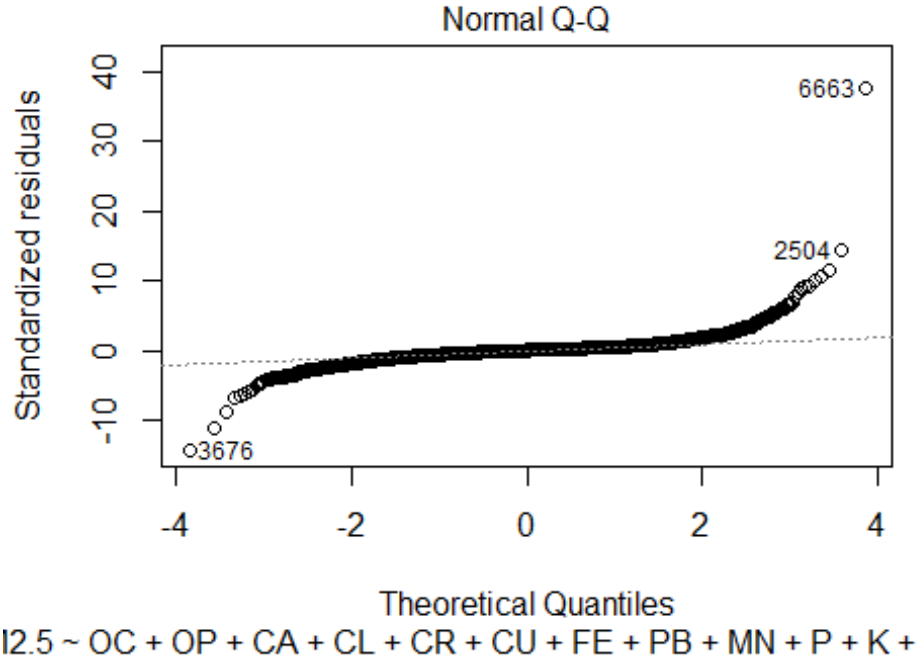


M2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P

```
plot(mod2, which = c(1,2))
```

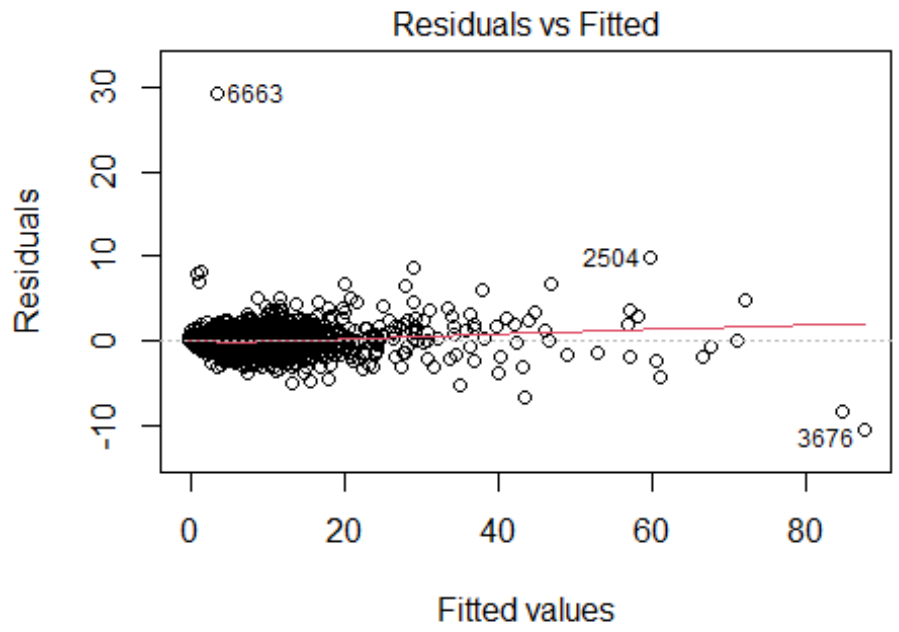


I2.5 ~ OC + OP + CA + CL + CR + CU + FE + PB + MN + P + K + RB

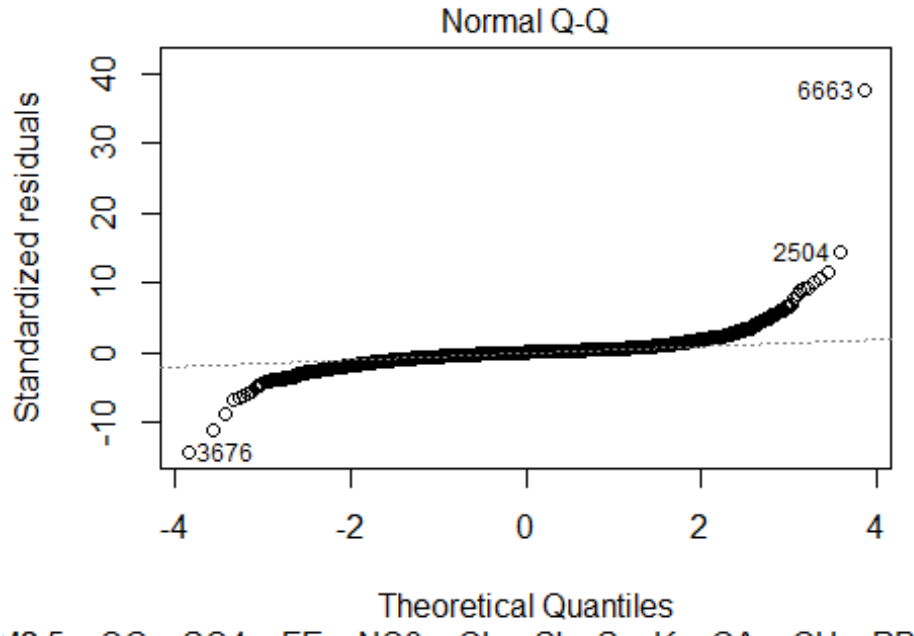


I2.5 ~ OC + OP + CA + CL + CR + CU + FE + PB + MN + P + K + RB

```
plot(mod3, which = c(1,2))
```

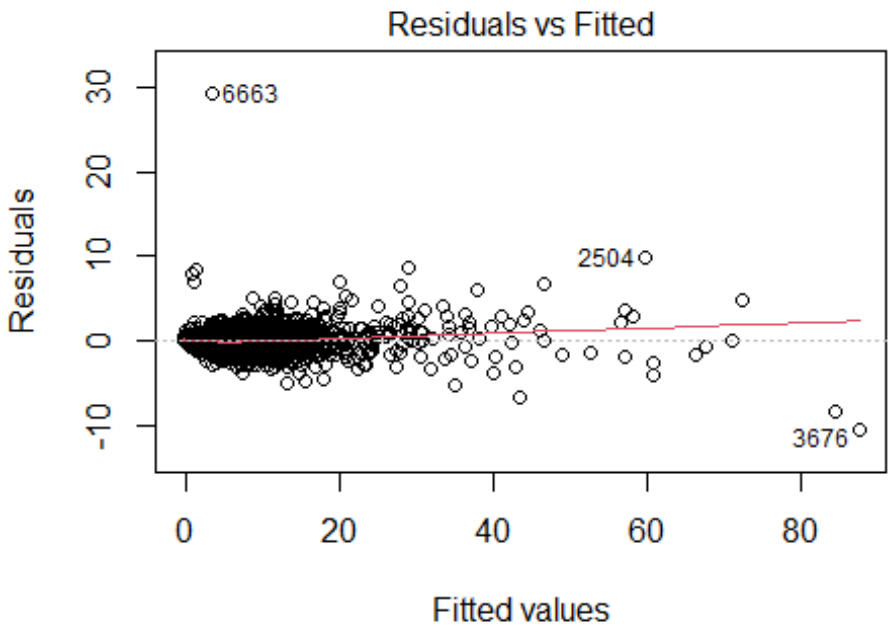


M2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P

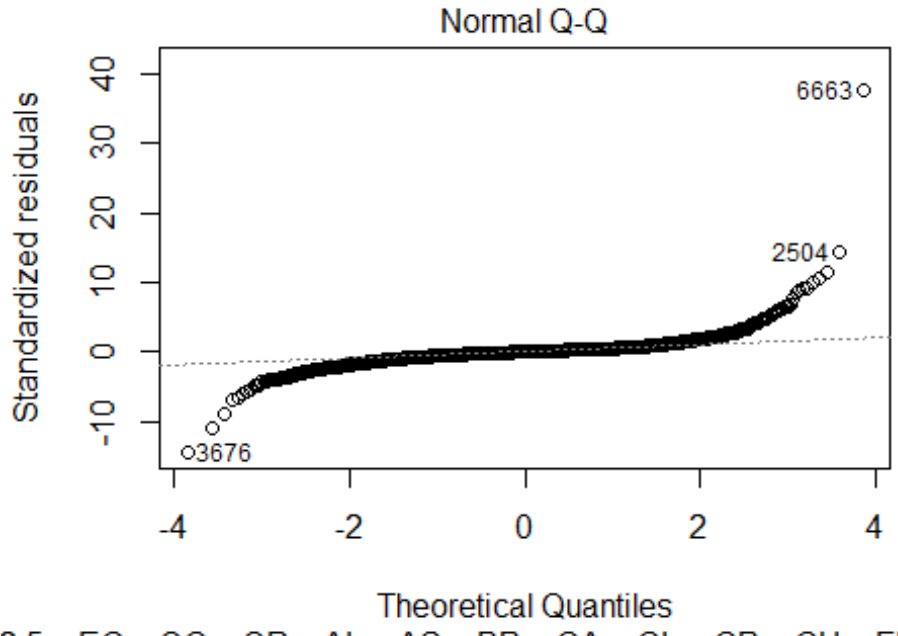


M2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P

```
plot(mod4, which = c(1,2))
```

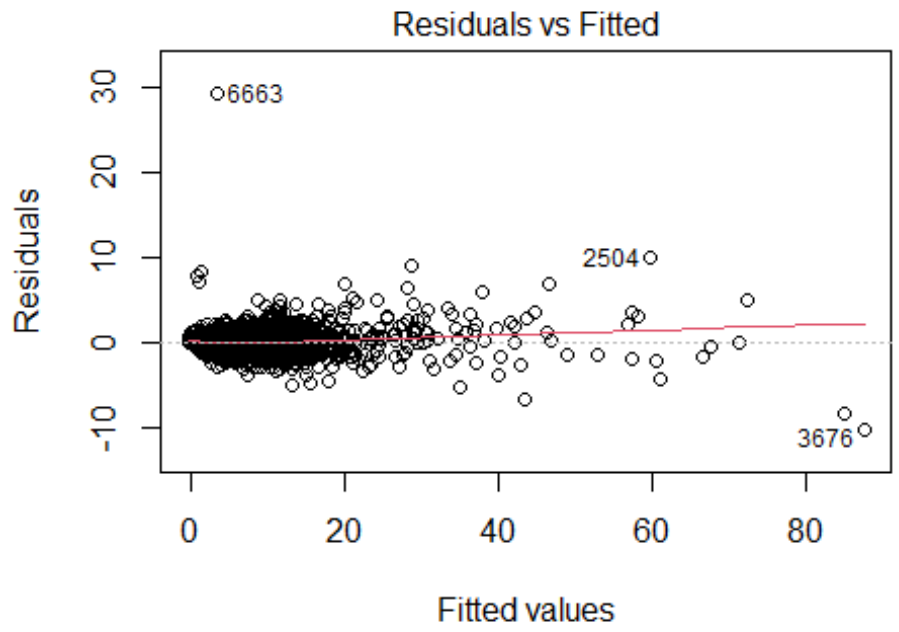


2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU + FE + P

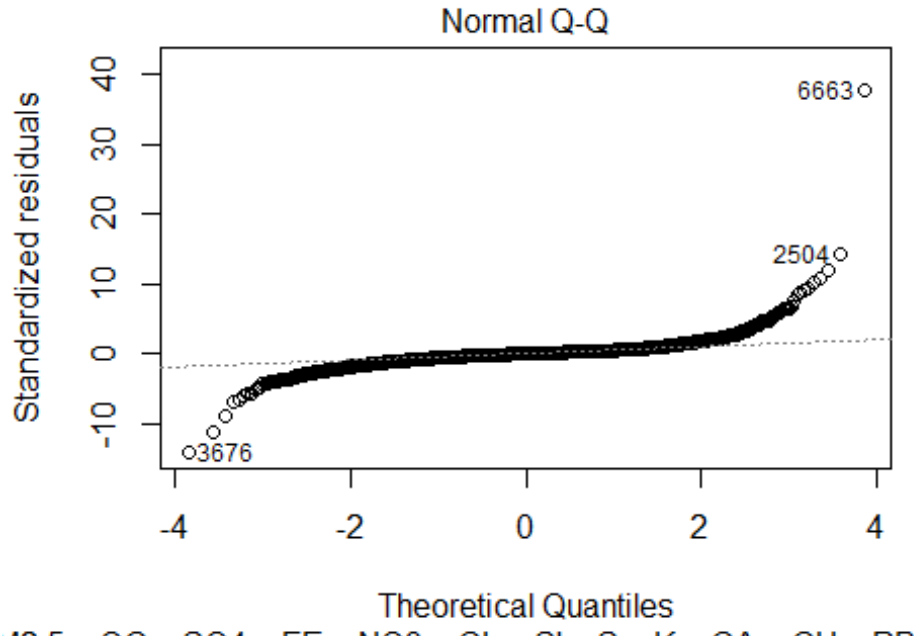


2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU + FE + P

```r
plot(mod5, which = c(1,2))
```

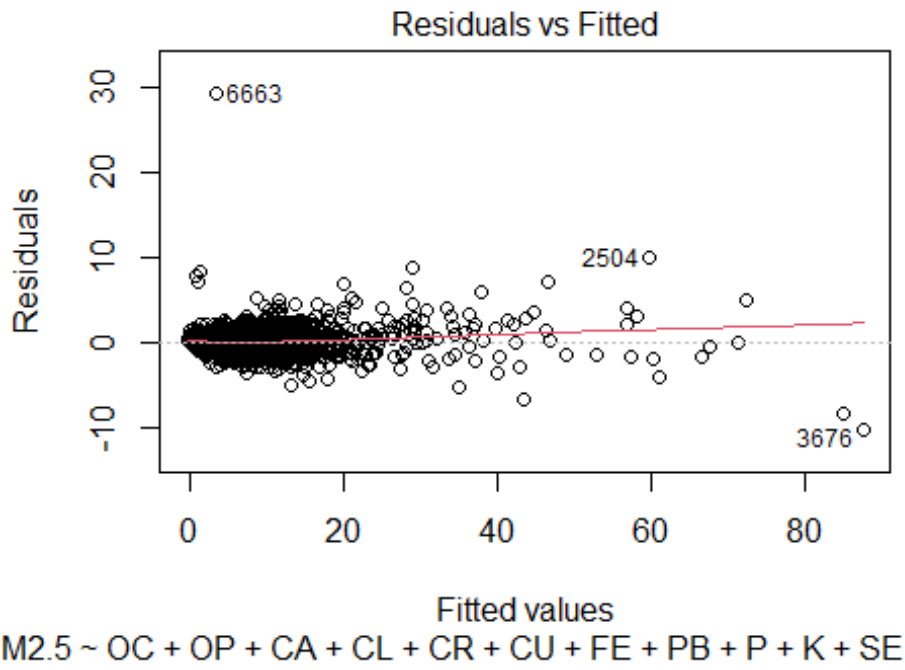


M2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P

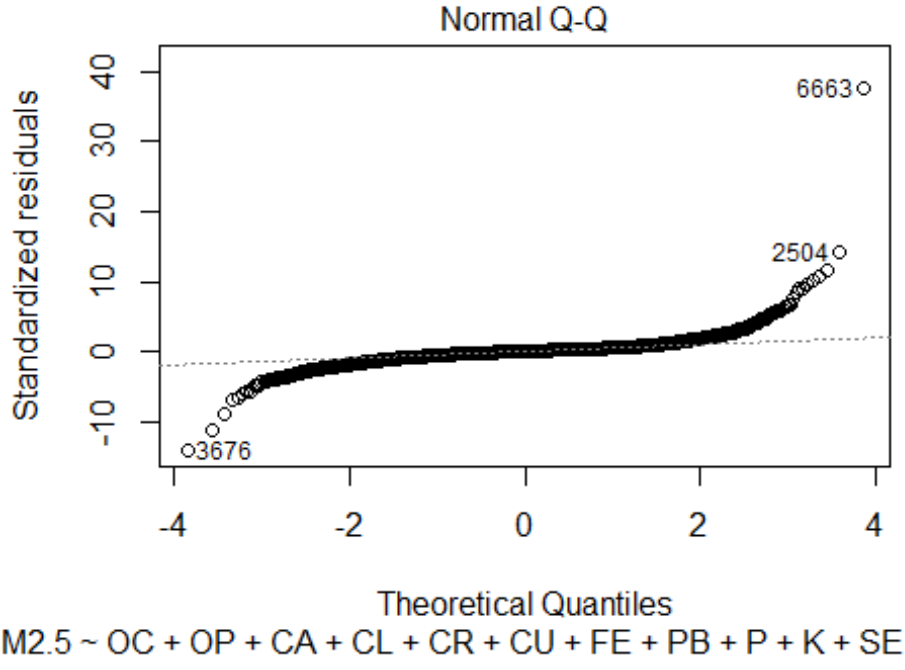


M2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P

```
plot(mod6, which = c(1,2))
```

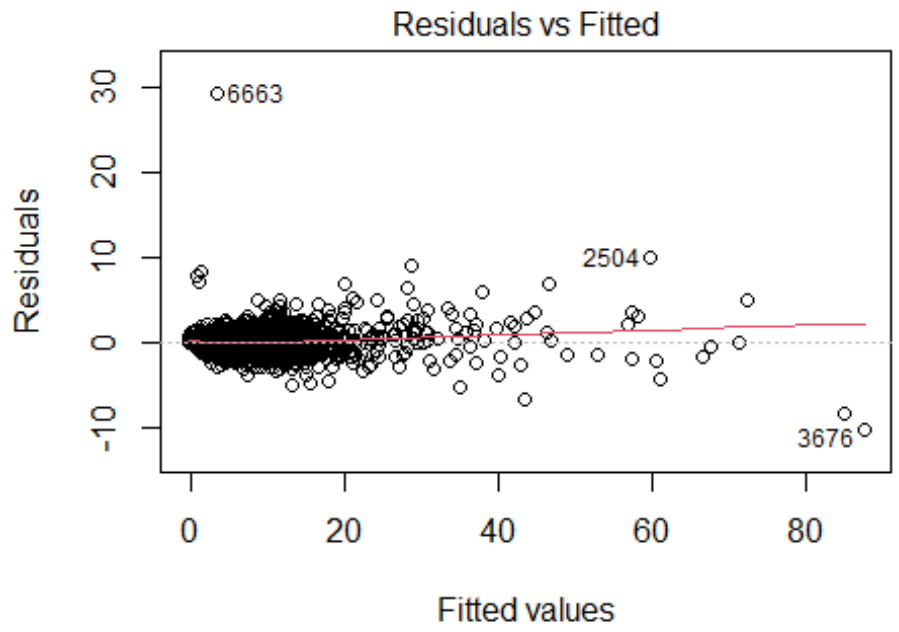


M2.5 ~ OC + OP + CA + CL + CR + CU + FE + PB + P + K + SE + SI

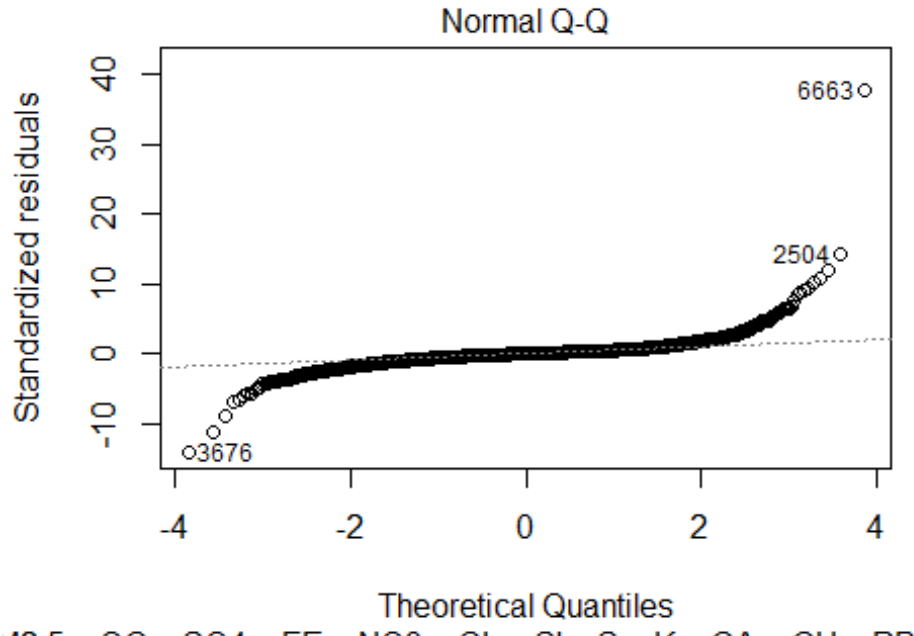


M2.5 ~ OC + OP + CA + CL + CR + CU + FE + PB + P + K + SE + SI

```
plot(mod7, which = c(1,2))
```

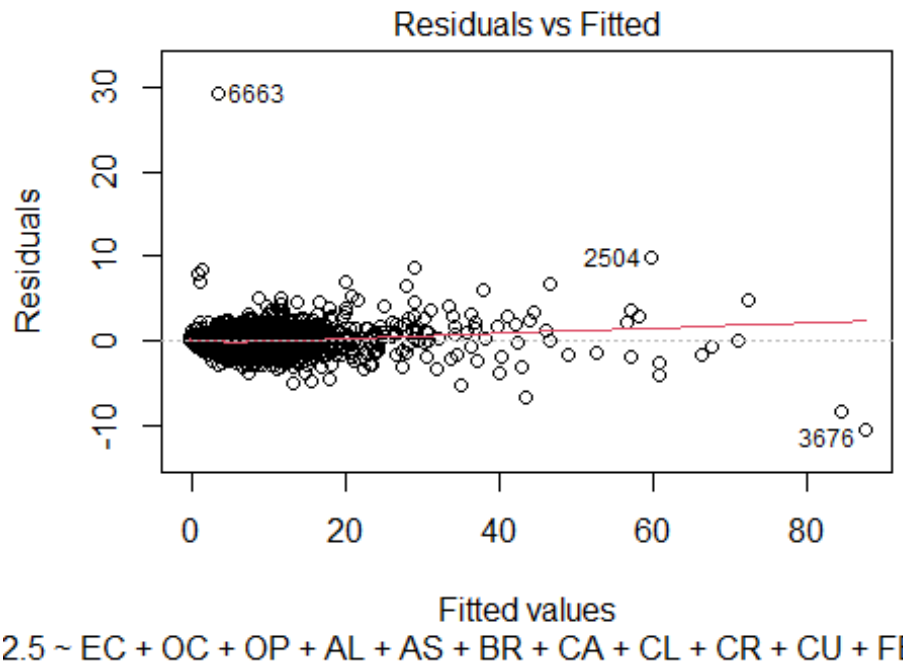


M2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P

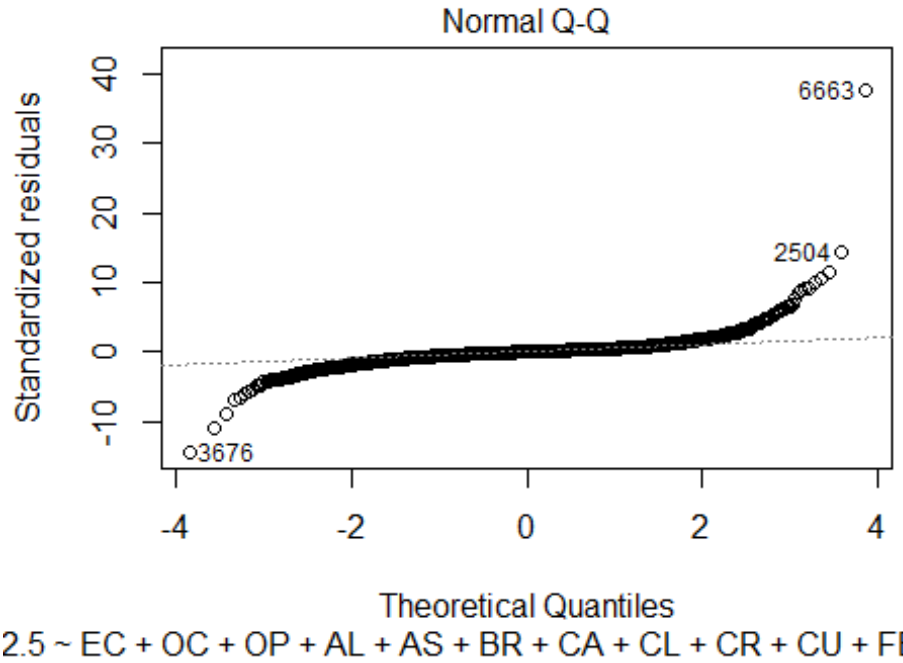


M2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P

```
plot(mod8, which = c(1,2))
```



2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU + FE + P



2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU + FE + P

```r
#model1
prediction = mod1 %>% predict(US_DATA_LRG_test)
data.frame( R2 = R2(prediction, US_DATA_LRG_test$PM2.5),
```

```
            RMSE = RMSE(prediction, US_DATA_LRG_test$PM2.5),
            MAE = MAE(prediction, US_DATA_LRG_test$PM2.5))

##          R2      RMSE      MAE
## 1 0.9780208 0.7720496 0.432098

#model2
prediction = mod2 %>% predict(US_DATA_LRG_test)
data.frame( R2 = R2(prediction, US_DATA_LRG_test$PM2.5),
            RMSE = RMSE(prediction, US_DATA_LRG_test$PM2.5),
            MAE = MAE(prediction, US_DATA_LRG_test$PM2.5))

##          R2      RMSE      MAE
## 1 0.9780208 0.7720496 0.432098

#model3
prediction = mod3 %>% predict(US_DATA_LRG_test)
data.frame( R2 = R2(prediction, US_DATA_LRG_test$PM2.5),
            RMSE = RMSE(prediction, US_DATA_LRG_test$PM2.5),
            MAE = MAE(prediction, US_DATA_LRG_test$PM2.5))

##          R2      RMSE      MAE
## 1 0.9780208 0.7720496 0.432098

#model4
prediction = mod4 %>% predict(US_DATA_LRG_test)
data.frame( R2 = R2(prediction, US_DATA_LRG_test$PM2.5),
            RMSE = RMSE(prediction, US_DATA_LRG_test$PM2.5),
            MAE = MAE(prediction, US_DATA_LRG_test$PM2.5))

##          R2      RMSE      MAE
## 1 0.9779738 0.7728304 0.432242

#model5
prediction = mod5 %>% predict(US_DATA_LRG_test)
data.frame( R2 = R2(prediction, US_DATA_LRG_test$PM2.5),
            RMSE = RMSE(prediction, US_DATA_LRG_test$PM2.5),
            MAE = MAE(prediction, US_DATA_LRG_test$PM2.5))

##          R2      RMSE       MAE
## 1 0.9780446 0.7717444 0.4316845

#model6
prediction = mod6 %>% predict(US_DATA_LRG_test)
data.frame( R2 = R2(prediction, US_DATA_LRG_test$PM2.5),
            RMSE = RMSE(prediction, US_DATA_LRG_test$PM2.5),
            MAE = MAE(prediction, US_DATA_LRG_test$PM2.5))

##          R2      RMSE      MAE
## 1 0.9780007 0.7724435 0.431843
```

```r
#model7
prediction = mod7 %>% predict(US_DATA_LRG_test)
data.frame( R2 = R2(prediction, US_DATA_LRG_test$PM2.5),
            RMSE = RMSE(prediction, US_DATA_LRG_test$PM2.5),
            MAE = MAE(prediction, US_DATA_LRG_test$PM2.5))

##          R2      RMSE       MAE
## 1 0.9780446 0.7717444 0.4316845

#model8
prediction = mod8 %>% predict(US_DATA_LRG_test)
data.frame( R2 = R2(prediction, US_DATA_LRG_test$PM2.5),
            RMSE = RMSE(prediction, US_DATA_LRG_test$PM2.5),
            MAE = MAE(prediction, US_DATA_LRG_test$PM2.5))

##          R2      RMSE      MAE
## 1 0.9779738 0.7728304 0.432242
```

#8 models were produced based on 8 different processes. They have similar adjusted coefficient of determination and their assumptions are valid. When testing their predictive ability, all of them have high R2 value and low Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) value.

```r
#consistency of regression coefficient
valid1 = lm(PM2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P
+ OP + TI + SE + V + CR + SR + MN + RB, data = US_DATA_LRG)
valid2 = lm(PM2.5 ~ OC + OP + CA + CL + CR + CU + FE + PB + MN + P + K + RB +
SE + SI + SR + S + TI + V + NO3 + SO4, data = US_DATA_LRG)
valid3 = lm(PM2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P
+ OP + TI + SE + V + CR + SR + MN + RB, data = US_DATA_LRG)
valid4 = lm(PM2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU + FE + PB
+ MG + MN + NI + N2 + P + K + RB + SE + SI + NA. + SR + S + TI + V + ZN + ZR
+ NO3 + SO4, data = US_DATA_LRG)
valid5 = lm(PM2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P
+ OP + TI + SE, data = US_DATA_LRG)
valid6 = lm(PM2.5 ~ OC + OP + CA + CL + CR + CU + FE + PB + P + K + SE + SI +
S + TI + V + NO3 + SO4, data = US_DATA_LRG)
valid7 = lm(PM2.5 ~ OC + SO4 + FE + NO3 + CL + SI + S + K + CA + CU + PB + P
+ OP + TI + SE, data = US_DATA_LRG)
valid8 = lm(PM2.5 ~ EC + OC + OP + AL + AS + BR + CA + CL + CR + CU + FE + PB
+ MG + MN + NI + N2 + P + K + RB + SE + SI + NA. + SR + S + TI + V + ZN + ZR
+ NO3 + SO4, data = US_DATA_LRG)
cbind(coef(summary(mod1))[,1], coef(summary(valid1))[,1])

##                    [,1]         [,2]
## (Intercept)   -0.2045776   -0.2045776
## OC             1.9231454    1.9231454
## SO4            0.3992574    0.3992574
## FE             3.7193158    3.7193158
## NO3            1.2209776    1.2209776
## CL             3.5368317    3.5368317
```

```
## SI             2.7651292    2.7651292
## S              3.9182363    3.9182363
## K              2.9583179    2.9583179
## CA             2.0211397    2.0211397
## CU           -26.0709656  -26.0709656
## PB            26.1187091   26.1187091
## P             45.1030623   45.1030623
## OP             0.2382850    0.2382850
## TI            15.0222095   15.0222095
## SE           146.7717588  146.7717588
## V             38.3276088   38.3276088
## CR          -154.9148578 -154.9148578
## SR           -15.1059510  -15.1059510
## MN           -22.3443123  -22.3443123
## RB            63.6468034   63.6468034
```

```
cbind(coef(summary(mod2))[,1], coef(summary(valid2))[,1])
```

```
##                    [,1]         [,2]
## (Intercept)   -0.2045776   -0.2045776
## OC             1.9231454    1.9231454
## OP             0.2382850    0.2382850
## CA             2.0211397    2.0211397
## CL             3.5368317    3.5368317
## CR          -154.9148578 -154.9148578
## CU           -26.0709656  -26.0709656
## FE             3.7193158    3.7193158
## PB            26.1187091   26.1187091
## MN           -22.3443123  -22.3443123
## P             45.1030623   45.1030623
## K              2.9583179    2.9583179
## RB            63.6468034   63.6468034
## SE           146.7717588  146.7717588
## SI             2.7651292    2.7651292
## SR           -15.1059510  -15.1059510
## S              3.9182363    3.9182363
## TI            15.0222095   15.0222095
## V             38.3276088   38.3276088
## NO3            1.2209776    1.2209776
## SO4            0.3992574    0.3992574
```

```
cbind(coef(summary(mod3))[,1], coef(summary(valid3))[,1])
```

```
##                    [,1]         [,2]
## (Intercept)   -0.2045776   -0.2045776
## OC             1.9231454    1.9231454
## SO4            0.3992574    0.3992574
## FE             3.7193158    3.7193158
## NO3            1.2209776    1.2209776
## CL             3.5368317    3.5368317
## SI             2.7651292    2.7651292
```

```
## S              3.9182363    3.9182363
## K              2.9583179    2.9583179
## CA             2.0211397    2.0211397
## CU           -26.0709656  -26.0709656
## PB            26.1187091   26.1187091
## P             45.1030623   45.1030623
## OP             0.2382850    0.2382850
## TI            15.0222095   15.0222095
## SE           146.7717588  146.7717588
## V             38.3276088   38.3276088
## CR          -154.9148578 -154.9148578
## SR           -15.1059510  -15.1059510
## MN           -22.3443123  -22.3443123
## RB            63.6468034   63.6468034

cbind(coef(summary(mod4))[,1], coef(summary(valid4))[,1])

##                      [,1]          [,2]
## (Intercept)   -0.21256462   -0.21256462
## EC            -0.11101415   -0.11101415
## OC             1.93475697    1.93475697
## OP             0.22447699    0.22447699
## AL            -0.58136572   -0.58136572
## AS            16.62335672   16.62335672
## BR             5.48740356    5.48740356
## CA             1.91323246    1.91323246
## CL             3.45763092    3.45763092
## CR          -148.03755914 -148.03755914
## CU           -26.39870378  -26.39870378
## FE             3.65516332    3.65516332
## PB            24.95061278   24.95061278
## MG            -0.03643026   -0.03643026
## MN           -20.57501447  -20.57501447
## NI            49.78919714   49.78919714
## N2             0.04224962    0.04224962
## P             44.97507876   44.97507876
## K              2.96531018    2.96531018
## RB            62.93135116   62.93135116
## SE           144.02217877  144.02217877
## SI             2.99261731    2.99261731
## NA.            0.11403627    0.11403627
## SR           -14.43223775  -14.43223775
## S              3.92477661    3.92477661
## TI            17.30420427   17.30420427
## V             25.86339643   25.86339643
## ZN            -0.55865933   -0.55865933
## ZR             9.27635580    9.27635580
## NO3            1.22060219    1.22060219
## SO4            0.39604787    0.39604787
```

```r
cbind(coef(summary(mod5))[,1], coef(summary(valid5))[,1])
```

```
##                    [,1]        [,2]
## (Intercept)  -0.2067957  -0.2067957
## OC            1.9301502   1.9301502
## SO4           0.4268946   0.4268946
## FE            2.2614638   2.2614638
## NO3           1.2238739   1.2238739
## CL            3.5397484   3.5397484
## SI            2.9902615   2.9902615
## S             3.8677503   3.8677503
## K             2.4362540   2.4362540
## CA            1.9113693   1.9113693
## CU          -29.5284818 -29.5284818
## PB           24.5011139  24.5011139
## P            47.1500977  47.1500977
## OP            0.2289468   0.2289468
## TI           18.4562834  18.4562834
## SE          141.3512728 141.3512728
```

```r
cbind(coef(summary(mod6))[,1], coef(summary(valid6))[,1])
```

```
##                     [,1]         [,2]
## (Intercept)   -0.2036429   -0.2036429
## OC             1.9290377    1.9290377
## OP             0.2314089    0.2314089
## CA             1.8704291    1.8704291
## CL             3.5338299    3.5338299
## CR          -170.0702343 -170.0702343
## CU           -25.8685233  -25.8685233
## FE             3.0005846    3.0005846
## PB            25.3082208   25.3082208
## P             45.4777712   45.4777712
## K              2.5654955    2.5654955
## SE           141.9082334  141.9082334
## SI             2.8900618    2.8900618
## S              3.8931129    3.8931129
## TI            15.8010448   15.8010448
## V             37.0164352   37.0164352
## NO3            1.2261903    1.2261903
## SO4            0.4097229    0.4097229
```

```r
cbind(coef(summary(mod7))[,1], coef(summary(valid7))[,1])
```

```
##                    [,1]        [,2]
## (Intercept)  -0.2067957  -0.2067957
## OC            1.9301502   1.9301502
## SO4           0.4268946   0.4268946
## FE            2.2614638   2.2614638
## NO3           1.2238739   1.2238739
## CL            3.5397484   3.5397484
```

```
## SI            2.9902615   2.9902615
## S             3.8677503   3.8677503
## K             2.4362540   2.4362540
## CA            1.9113693   1.9113693
## CU          -29.5284818 -29.5284818
## PB           24.5011139  24.5011139
## P            47.1500977  47.1500977
## OP            0.2289468   0.2289468
## TI           18.4562834  18.4562834
## SE          141.3512728 141.3512728
```

```
cbind(coef(summary(mod8))[,1], coef(summary(valid8))[,1])
```

```
##                      [,1]          [,2]
## (Intercept)   -0.21256462   -0.21256462
## EC            -0.11101415   -0.11101415
## OC             1.93475697    1.93475697
## OP             0.22447699    0.22447699
## AL            -0.58136572   -0.58136572
## AS            16.62335672   16.62335672
## BR             5.48740356    5.48740356
## CA             1.91323246    1.91323246
## CL             3.45763092    3.45763092
## CR          -148.03755914 -148.03755914
## CU           -26.39870378  -26.39870378
## FE             3.65516332    3.65516332
## PB            24.95061278   24.95061278
## MG            -0.03643026   -0.03643026
## MN           -20.57501447  -20.57501447
## NI            49.78919714   49.78919714
## N2             0.04224962    0.04224962
## P             44.97507876   44.97507876
## K              2.96531018    2.96531018
## RB            62.93135116   62.93135116
## SE           144.02217877  144.02217877
## SI             2.99261731    2.99261731
## NA.            0.11403627    0.11403627
## SR           -14.43223775  -14.43223775
## S              3.92477661    3.92477661
## TI            17.30420427   17.30420427
## V             25.86339643   25.86339643
## ZN            -0.55865933   -0.55865933
## ZR             9.27635580    9.27635580
## NO3            1.22060219    1.22060219
## SO4            0.39604787    0.39604787
```

#The regression coefficients are consistency between training data and testing data in all of these models.

```
#Complexity of models
length(coef(summary(mod1))[,1])
```

```
## [1] 21
```

```
length(coef(summary(mod2))[,1])
```

```
## [1] 21
```

```
length(coef(summary(mod3))[,1])
```

```
## [1] 21
```

```
length(coef(summary(mod4))[,1])
```

```
## [1] 31
```

```
length(coef(summary(mod5))[,1])
```

```
## [1] 16
```

```
length(coef(summary(mod6))[,1])
```

```
## [1] 18
```

```
length(coef(summary(mod7))[,1])
```

```
## [1] 16
```

```
length(coef(summary(mod8))[,1])
```

```
## [1] 31
```

STA 141A Project (ENR)

Seyoung Jung

12/15/2020

— Step 1: Data loading and procressing —

```
## --- Part a: Upload Metadata for samples ---
setwd("C:/Users/Martin/Desktop/Fall 2020/STA 141A")
path_data<-file.path(getwd(),"Project")
META_DATA<-as_tibble(read.csv(file.path(path_data,"IMPROVE_metadata.csv")))
## --- Filter samples from Korea and Canada ---
US_META<-META_DATA %>% filter(Country %nin% c("KR","CA"))
## --- Filter stats not in continental US ---
US_META<-META_DATA %>% filter(State %nin% c("HI","AK","VI"))
## --- Part b: Load samples data ---
DATA<-as_tibble(read.csv(file.path(path_data,"IMPROVE_2015_data_w_UNC_v2.csv")))
## --- Part c: Select samples from SW given site identifiers from SW_META table ("C
ode")
US_DATA_all<-as_tibble(DATA %>% filter(SiteCode %in% US_META$Code))
```

```
# Let's identify any samples that (grossly) violate PM2.5 mass balances
# PM2.5 (=Y) cannot be negative!
# Since there's some probability that PM2.5 is negative due to errors at low concen
tration, we may use PM2.5 uncertainties to remove samples that fall outside -3*PM2.
5_UNC.
# In this way, we don't risk censoring the data but do remove likely erroneous dat
a.
US_DATA_all<-US_DATA_all %>% dplyr::filter(PM2.5 > -3*PM2.5_UNC)
```

```
exclude<-c("fAbs","PM10","POC","ammNO3","ammSO4","SOIL","SeaSalt","OC1","OC2","OC
3","OC4","EC1","EC2","EC3","fAbs_MDL")
US_DATA_LRG<- US_DATA_all %>% dplyr::select(!contains(exclude) & !matches("_UNC") |
matches("PM2.5_UNC"))
any(is.na(US_DATA_LRG))
```

```
## [1] TRUE
```

```
US_DATA_LRG<-US_DATA_LRG[which(complete.cases(US_DATA_LRG)),]
any(is.na(US_DATA_LRG))
```

```
## [1] FALSE
```

```
## --- Instead of random partitioning, I will partition by first sorting samples by
SiteCode and DATE (already done) and place every other sample in the test set.
# --- This data has seasonality. Sorting by date therefore ensures seasonality is e
quivalent between datasets
n<-nrow(US_DATA_LRG)
ind_test<-seq(1,n,2)
US_DATA_LRG_test<-US_DATA_LRG[ind_test,]
US_DATA_LRG<-US_DATA_LRG[-ind_test,]
```

Categorical => dummy Test

Two of our predictor variables are categorical variables (SiteCode and Date). Hence, we need to convert the variables to dummy variables. Also, in order to use the cv.glmnet function to fit the Elastic Net Regression to the data, input data should be in a matrix format. Also, since the function does not accept formula notation, x and y must be passed in separately. So, we create two different sets for training set.

```
US_DATA_LRG_train_y <- US_DATA_LRG$PM2.5
x_train_cont <- US_DATA_LRG %>%
  select(-PM2.5, -SiteCode, -Date, -PM2.5_UNC) %>%
  as.matrix()
x_train_cat <- US_DATA_LRG %>%
  select(SiteCode, Date) %>%
  model.matrix( ~ .-1, .)
US_DATA_LRG_train_x <- cbind(x_train_cont, x_train_cat)


US_DATA_LRG_test_y <- US_DATA_LRG_test$PM2.5
x_test_cont <- US_DATA_LRG_test %>%
  select(-PM2.5, -SiteCode, -Date, -PM2.5_UNC) %>%
  as.matrix()
x_test_cat <- US_DATA_LRG_test %>%
  select(SiteCode, Date) %>%
  model.matrix( ~ .-1, .)
US_DATA_LRG_test_x <- cbind(x_test_cont, x_test_cat)
```

After converting the factor variables, the training set (US_DATA_LRG_train_x) will have 308 variables.

Now, we will fit models to the training data. By default, the cv.glmnet uses 10-fold cross validation to find the optimal values for lambda. Also, we will use the mean squared error for our evaluation metric. When alpha is 0 (or 1), this function fits Ridge Regression (or Lasso Regression). We will try 20 different values, between 0 and 1, for alpha to find a value that gives us the best result.

```
set.seed(141)

fits_list <- list()
for (i in 0:20) {
  fits_name <- paste0("Alpha_", i/20)

  fits_list[[fits_name]] <- cv.glmnet(as.matrix(US_DATA_LRG_train_x), as.matrix(US_
DATA_LRG_train_y), type.measure="mse", alpha=i/20, family="gaussian")
}
```

lambda.1se is the value for lambda, stored in each fitted model, that resulted in the simplest model (i.e. the model with the least non-zero parameters) and was within 1 standard error of the lambda that had the smallest sum.

```
fit_result <- data.frame()
for (i in 0:20) {
  fits_name <- paste0("Alpha_", i/20)

  predict_val <- predict(fits_list[[fits_name]], s=fits_list[[fits_name]]$lambda.1s
e, newx=as.matrix(US_DATA_LRG_test_x))

  mse <- mean((as.matrix(US_DATA_LRG_test_y) - predict_val)^2)

  temp_val <- data.frame(Alpha=i/20, MSE=mse, fits_name=fits_name)
  fit_result <- rbind(fit_result, temp_val)
  }

fit_result
```

```
##    Alpha       MSE  fits_name
## 1   0.00 0.7634257    Alpha_0
## 2   0.05 0.6054706 Alpha_0.05
## 3   0.10 0.6065652  Alpha_0.1
## 4   0.15 0.6123570 Alpha_0.15
## 5   0.20 0.6071487  Alpha_0.2
## 6   0.25 0.6166402 Alpha_0.25
## 7   0.30 0.6084149  Alpha_0.3
## 8   0.35 0.6113124 Alpha_0.35
## 9   0.40 0.6070638  Alpha_0.4
## 10  0.45 0.5977094 Alpha_0.45
## 11  0.50 0.6352535  Alpha_0.5
## 12  0.55 0.6141631 Alpha_0.55
## 13  0.60 0.6186419  Alpha_0.6
## 14  0.65 0.6227428 Alpha_0.65
## 15  0.70 0.6171354  Alpha_0.7
## 16  0.75 0.6373969 Alpha_0.75
## 17  0.80 0.6155697  Alpha_0.8
## 18  0.85 0.6355489 Alpha_0.85
## 19  0.90 0.6405099  Alpha_0.9
## 20  0.95 0.6545169 Alpha_0.95
## 21  1.00 0.6332386    Alpha_1
```

We can see that neither Ridge Regression nor Lasso Regression gives us the best result. Although it gives us a very similar MSE values when alpha is in between 0 and 1, it has the lowest MSE when alpha=0.45. Since we are using Elastic Net Regression, we can expect that this model has less predictor variables than the full model.

For the model with alpha=0.45, cross validation method chooses lambda=0.074. If we take a closer look at a model with alpha=0.45, we can observe that 52 of variables are nonzero. It means that the remaining variables are dropped when fitting this model to the training set.

```
# We can see that the mse is the lowest when alpha = 0.45.
fits_list$Alpha_0.45
```

```
##
## Call:  cv.glmnet(x = as.matrix(US_DATA_LRG_train_x), y = as.matrix(US_DATA_LRG_t
rain_y),      type.measure = "mse", alpha = i/20, family = "gaussian")
##
## Measure: Mean-Squared Error
##
##      Lambda Measure      SE Nonzero
## min 0.00723  0.5523 0.08607     261
## 1se 0.07397  0.6359 0.08232      52
```

```
fits_list$Alpha_0.45$lambda.1se # value for lambda
```

```
## [1] 0.07397486
```

```
coef(fits_list$Alpha_0.45)      # coefficients of this model
```

```
## 309 x 1 sparse Matrix of class "dgCMatrix"
##                              1
## (Intercept)     -0.114674053
## EC               0.426403466
## OC               1.711304246
## OP               0.807417478
## AL               0.971613825
## AS               .
## BR              12.595389511
## CA               1.756914687
## CL               2.745861899
## CR               .
## CU               .
## FE               3.510792738
## PB               8.392527873
## MG               1.062569693
## MN               .
## NI               .
## N2               .
## P               27.117590226
## K                1.768529297
## RB              85.508506989
## SE             127.200123974
## SI               2.285695803
## NA.              0.362930418
## SR               .
## S                3.102419694
## TI               9.062758625
## V               18.565709821
## ZN               .
## ZR               .
## NO3              1.178011209
## SO4              0.618401342
## SiteCodeACAD1    .
## SiteCodeAGTI1    .
## SiteCodeBADL1    .
## SiteCodeBALA1    .
## SiteCodeBALD1    .
## SiteCodeBAND1    .
## SiteCodeBIBE1    .
## SiteCodeBIRM1    .
## SiteCodeBLIS1    .
## SiteCodeBLMO1    .
## SiteCodeBOAP1    .
## SiteCodeBOLA1    .
## SiteCodeBOND1    .
## SiteCodeBOWA1    .
## SiteCodeBRCA1    .
## SiteCodeBRID1    .
## SiteCodeBRIG1    .
```

```
## SiteCodeBRIS1    .
## SiteCodeBRMA1    .
## SiteCodeBYIS1    .
## SiteCodeCABA1    .
## SiteCodeCABI1    .
## SiteCodeCACO1    .
## SiteCodeCACR1    .
## SiteCodeCANY1    .
## SiteCodeCAPI1    .
## SiteCodeCEBL1    .
## SiteCodeCHAS1    .
## SiteCodeCHIR1    .
## SiteCodeCLPE1    .
## SiteCodeCOHU1    .
## SiteCodeCORI1   -0.010450120
## SiteCodeCRES1    .
## SiteCodeCRLA1    .
## SiteCodeCRMO1    .
## SiteCodeDOME1    .
## SiteCodeDOSO1    .
## SiteCodeDOUG1    .
## SiteCodeEGBE1    .
## SiteCodeELDO1   -0.136389716
## SiteCodeELLI1    .
## SiteCodeEVER1    .
## SiteCodeFLAT1    .
## SiteCodeFLTO1    .
## SiteCodeFOPE1    .
## SiteCodeFRES1    .
## SiteCodeFRRE1    .
## SiteCodeGAMO1    .
## SiteCodeGICL1    .
## SiteCodeGLAC1    .
## SiteCodeGRBA1    .
## SiteCodeGRCA2    .
## SiteCodeGRGU1    .
## SiteCodeGRRI1    .
## SiteCodeGRSA1    .
## SiteCodeGRSM1    .
## SiteCodeGUMO1    .
## SiteCodeHECA1    .
## SiteCodeHEGL1    .
## SiteCodeHOOV1    .
## SiteCodeIKBA1    .
## SiteCodeISLE1    .
## SiteCodeJARB1    .
## SiteCodeJARI1    .
## SiteCodeJOSH1    .
## SiteCodeKAIS1    .
## SiteCodeKALM1    .
## SiteCodeLABE1    .
```

```
## SiteCodeLASU2    .
## SiteCodeLAVO1    .
## SiteCodeLIGO1    .
## SiteCodeLOND1    .
## SiteCodeLOST1    .
## SiteCodeLTCC1    .
## SiteCodeLYEB1    .
## SiteCodeMACA1    .
## SiteCodeMAKA2    0.002197642
## SiteCodeMAVI1    0.088369079
## SiteCodeMEAD1    .
## SiteCodeMELA1    .
## SiteCodeMEVE1    .
## SiteCodeMING1    .
## SiteCodeMOHO1    .
## SiteCodeMOMO1    .
## SiteCodeMONT1    .
## SiteCodeMOOS1    .
## SiteCodeMORA1    .
## SiteCodeMOZI1    .
## SiteCodeNEBR1    .
## SiteCodeNOAB1    .
## SiteCodeNOCA1    .
## SiteCodeNOCH1    .
## SiteCodeNOGA1    .
## SiteCodeOKEF1    .
## SiteCodeOLYM1    .
## SiteCodeORPI1    .
## SiteCodeOWVL1    .
## SiteCodePACK1    .
## SiteCodePASA1    .
## SiteCodePEFO1    .
## SiteCodePENO1    .
## SiteCodePHOE1   -0.127508792
## SiteCodePHOE5    .
## SiteCodePINN1    .
## SiteCodePMRF1    .
## SiteCodePORE1    0.660387523
## SiteCodePRIS1    .
## SiteCodePUSO1    .
## SiteCodeQUCI1    .
## SiteCodeQURE1    .
## SiteCodeQUVA1    .
## SiteCodeRAFA1    .
## SiteCodeREDW1    0.017723498
## SiteCodeROMA1    .
## SiteCodeROMO1    .
## SiteCodeSACR1    .
## SiteCodeSAGA1   -0.103433378
## SiteCodeSAGO1    .
## SiteCodeSAGU1    .
```

```
## SiteCodeSAMA1    0.074553728
## SiteCodeSAPE1    .
## SiteCodeSAWE1    .
## SiteCodeSAWT1    .
## SiteCodeSENE1    .
## SiteCodeSEQU1    .
## SiteCodeSHEN1    .
## SiteCodeSHMI1    .
## SiteCodeSHRO1    .
## SiteCodeSIAN1    .
## SiteCodeSIPS1    .
## SiteCodeSNPA1    .
## SiteCodeSTAR1    .
## SiteCodeSTIL1    .
## SiteCodeSULA1    .
## SiteCodeSWAN1    .
## SiteCodeSYCA1    .
## SiteCodeSYCA2    .
## SiteCodeTALL1    .
## SiteCodeTHBA1    .
## SiteCodeTHRO1    .
## SiteCodeTHSI1    .
## SiteCodeTONT1    .
## SiteCodeTRIN1    .
## SiteCodeULBE1    .
## SiteCodeUPBU1    .
## SiteCodeVILA1    .
## SiteCodeVOYA2    .
## SiteCodeWASH1    .
## SiteCodeWEMI1    .
## SiteCodeWHIT1    .
## SiteCodeWHPA1    .
## SiteCodeWHPE1    .
## SiteCodeWHRI1    .
## SiteCodeWICA1    .
## SiteCodeWIMO1    .
## SiteCodeYELL2    .
## SiteCodeYOSE1    .
## SiteCodeZICA1    .
## Date1/15/2015    .
## Date1/18/2015    .
## Date1/21/2015    .
## Date1/24/2015    .
## Date1/27/2015    .
## Date1/3/2015     .
## Date1/30/2015    .
## Date1/6/2015     .
## Date1/9/2015     .
## Date10/12/2015   .
## Date10/15/2015   .
## Date10/18/2015   .
```

```
## Date10/21/2015   .
## Date10/24/2015   .
## Date10/27/2015   .
## Date10/3/2015    .
## Date10/30/2015   .
## Date10/6/2015    .
## Date10/9/2015    .
## Date11/11/2015   .
## Date11/14/2015   .
## Date11/17/2015   .
## Date11/2/2015    .
## Date11/20/2015   .
## Date11/23/2015   .
## Date11/26/2015   .
## Date11/29/2015   .
## Date11/5/2015    .
## Date11/8/2015    .
## Date12/11/2015   .
## Date12/14/2015   .
## Date12/17/2015   .
## Date12/2/2015    .
## Date12/20/2015   .
## Date12/23/2015   .
## Date12/26/2015   .
## Date12/29/2015   .
## Date12/5/2015    .
## Date12/8/2015    .
## Date2/11/2015    .
## Date2/14/2015    .
## Date2/17/2015    .
## Date2/2/2015     .
## Date2/20/2015    .
## Date2/23/2015    .
## Date2/26/2015    .
## Date2/5/2015     .
## Date2/8/2015     .
## Date3/1/2015     .
## Date3/10/2015    .
## Date3/13/2015   -0.011986774
## Date3/16/2015    .
## Date3/19/2015   -0.026328360
## Date3/22/2015   -0.076507845
## Date3/25/2015    .
## Date3/28/2015    .
## Date3/31/2015    .
## Date3/4/2015    -0.009827235
## Date3/7/2015    -0.258708631
## Date4/12/2015    .
## Date4/15/2015    .
## Date4/18/2015    .
## Date4/21/2015    .
```

```
## Date4/24/2015    .
## Date4/27/2015    .
## Date4/3/2015     .
## Date4/30/2015    .
## Date4/6/2015     .
## Date4/9/2015     .
## Date5/12/2015    .
## Date5/15/2015    .
## Date5/18/2015    .
## Date5/21/2015    .
## Date5/24/2015    .
## Date5/27/2015    .
## Date5/3/2015     .
## Date5/30/2015    .
## Date5/6/2015     .
## Date5/9/2015     .
## Date6/11/2015    0.284272033
## Date6/14/2015    .
## Date6/17/2015    .
## Date6/2/2015     .
## Date6/20/2015    .
## Date6/23/2015    0.044890657
## Date6/26/2015    .
## Date6/29/2015    0.054805817
## Date6/5/2015     .
## Date6/8/2015     .
## Date7/11/2015    .
## Date7/14/2015    .
## Date7/17/2015    .
## Date7/2/2015     0.318254665
## Date7/20/2015    .
## Date7/23/2015    .
## Date7/26/2015    .
## Date7/29/2015    0.186479582
## Date7/5/2015     0.026638887
## Date7/8/2015     .
## Date8/1/2015     .
## Date8/10/2015    0.003149496
## Date8/13/2015    0.003635989
## Date8/16/2015    0.157029484
## Date8/19/2015    0.288633654
## Date8/22/2015    .
## Date8/25/2015    0.227724937
## Date8/28/2015    0.222549734
## Date8/31/2015    0.123963928
## Date8/4/2015     0.195171963
## Date8/7/2015     0.147065416
## Date9/12/2015    .
## Date9/15/2015    .
## Date9/18/2015    0.027475458
## Date9/21/2015    .
```

```
## Date9/24/2015    .
## Date9/27/2015    .
## Date9/3/2015     0.091789732
## Date9/30/2015    .
## Date9/6/2015     .
## Date9/9/2015     .
```

STA141A-Tree Models-ATW-CMD-Markdown

Andrew T. Weakley, Christina De Cesaris

12/15/2020

— Step 1: Data loading and procressing —

```
## --- Part a: Upload Metadata for samples ---
path_data<-file.path(getwd(),"data")
META_DATA<-as_tibble(read.csv(file.path(path_data,"IMPROVE_metadata.csv")))
## --- Filter samples from Korea and Canada ---
US_META<-META_DATA %>% filter(Country %nin% c("KR","CA"))


## --- Filter stats not in continental US ---
US_META<-META_DATA %>% filter(State %nin% c("HI","AK","VI"))

## --- Part b: Load samples data ---
DATA<-as_tibble(read.csv(file.path(path_data,"IMPROVE_2015_data_w_UNC_v2.csv")))

## --- Part c: Select samples from SW given site identifiers from SW_META table ("C
ode")
US_DATA_all<-as_tibble(DATA %>% filter(SiteCode %in% US_META$Code))
```

```
# Let's identify any samples that (grossly) violate PM2.5 mass balances
# PM2.5 (=Y) cannot be negative!
# Since there's some probability that PM2.5 is negative due to errors at low concen
tration, we may use PM2.5 uncertainties to remove samples that fall outside -3*PM2.
5_UNC.
# In this way, we don't risk censoring the data but do remove likely erroneous dat
a.
US_DATA_all<-US_DATA_all %>% dplyr::filter(PM2.5 > -3*PM2.5_UNC)
```

```
exclude<-c("PM10","POC","ammNO3","ammSO4","SOIL","SeaSalt","OC1","OC2","OC3","OC
4","EC1","EC2","EC3","fAbs_MDL","fAbs")
US_DATA_LRG<- US_DATA_all %>% dplyr::select(!contains(exclude) & !matches("_UNC") |
matches("PM2.5_UNC"))
any(is.na(US_DATA_LRG))
```

```
## [1] TRUE
```

```
US_DATA_LRG<-US_DATA_LRG[which(complete.cases(US_DATA_LRG)),]
any(is.na(US_DATA_LRG))
```

```
## [1] FALSE
```

```
set.seed(123)
## --- Instead of random partitioning, I will partition by first sorting samples by
SiteCode and DATE (already done) and place every other sample in the test set.
# --- This data has seasonality. Sorting by date therefore ensures seasonality is e
quivalent between datasets
n<-nrow(US_DATA_LRG)
ind_test<-seq(1,n,2)
US_DATA_LRG_test<-US_DATA_LRG[ind_test,]
US_DATA_LRG<-US_DATA_LRG[-ind_test,]
```

— Step 2: mclust for GMMs —

```
## --- Normalize US data by PM2.5 conc --
US_DATA_LRG_PM_norm<-US_DATA_LRG %>% dplyr::select(everything()/"PM2.5")
#rename_with()
```

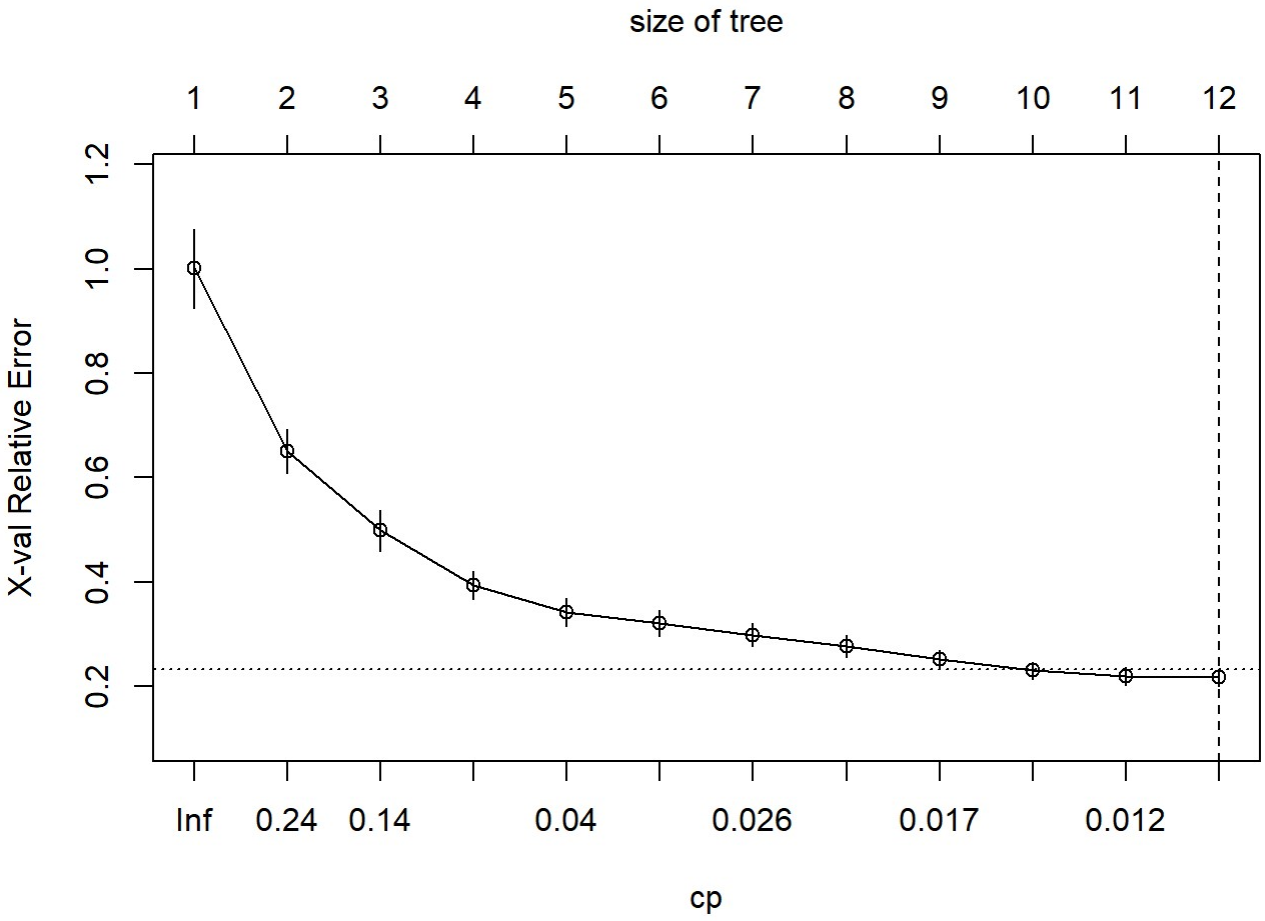
— Tree Regression —

1. initial fits

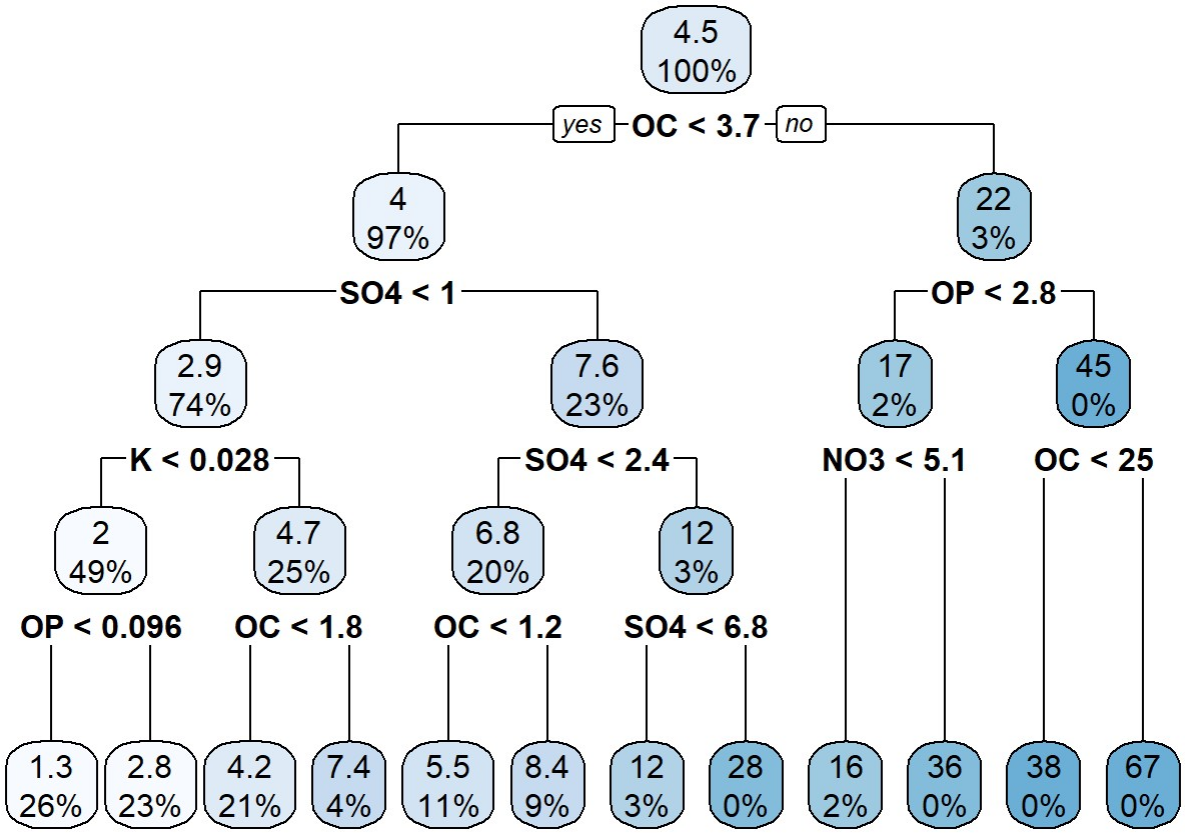
```
fit1 <- rpart(
    formula = PM2.5 ~ .-PM2.5_UNC,
    data    = US_DATA_LRG,
    method  = "anova",
    control = list(minsplit = 10, maxdepth = 20, xval = 10)
)
fit1
```

```
## n= 8647
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 8647 219000.9000  4.549602
##    2) OC< 3.693035 8390  90149.2100  4.017542
##      4) SO4< 1.0019 6378  26722.0200  2.890826
##        8) K< 0.02811 4237   6138.1860  1.992608
##         16) OP< 0.09625 2207   1385.3870  1.256258 *
##         17) OP>=0.09625 2030   2255.1420  2.793161 *
##        9) K>=0.02811 2141  10400.5200  4.668382
##         18) OC< 1.81129 1819   6084.1310  4.182025 *
##         19) OC>=1.81129 322   1455.4710  7.415849 *
##      5) SO4>=1.0019 2012  29663.6600  7.589211
##       10) SO4< 2.38635 1728  13878.1800  6.802017
##         20) OC< 1.16163 964   4246.3170  5.500812 *
##         21) OC>=1.16163 764   5940.2230  8.443853 *
##       11) SO4>=2.38635 284   8199.4220 12.378890
##         22) SO4< 6.76665 270   3368.5850 11.546900 *
##         23) SO4>=6.76665 14   1039.4700 28.424520 *
##    3) OC>=3.693035 257  48939.1500 21.919180
##      6) OP< 2.8095 214  12843.1400 17.344850
##       12) NO3< 5.11497 201   5742.5780 16.107340 *
##       13) NO3>=5.11497 13   2033.4390 36.478600 *
##      7) OP>=2.8095 43   9333.0300 44.684480
##       14) OC< 25.39485 33   2434.7920 38.052000 *
##       15) OC>=25.39485 10    656.0985 66.571640 *
```

```
#pairs(US_DATA_LRG[which(sapply(US_DATA_LRG, is.numeric))])
plotcp(fit1)

abline(v = 12, lty = "dashed")
```




```
rpart.plot(fit1)
```




```
summary(fit1)
```

```
## Call:
## rpart(formula = PM2.5 ~ . - PM2.5_UNC, data = US_DATA_LRG, method = "anova",
##     control = list(minsplit = 10, maxdepth = 20, xval = 10))
##   n= 8647
##
##            CP nsplit rel error    xerror       xstd
## 1  0.36489610      0 1.0000000 1.0003518 0.07605944
## 2  0.15417066      1 0.6351039 0.6501399 0.04173009
## 3  0.12220485      2 0.4809332 0.4985611 0.03998255
## 4  0.04649899      3 0.3587284 0.3941103 0.02711683
## 5  0.03463939      4 0.3122294 0.3421852 0.02628566
## 6  0.02850279      5 0.2775900 0.3213551 0.02532643
## 7  0.02313744      6 0.2490872 0.2992325 0.02255095
## 8  0.01731210      7 0.2259498 0.2773869 0.02106606
## 9  0.01685673      8 0.2086377 0.2520433 0.01790144
## 10 0.01306348      9 0.1917809 0.2305255 0.01659266
## 11 0.01140478     10 0.1787175 0.2197864 0.01640755
## 12 0.01000000     11 0.1673127 0.2171949 0.01646286
##
## Variable importance
##       OC       OP      SO4        S       EC SiteCode        K       SE
##       28       21       11       10        7        5        3        2
##       ZN      NO3     Date       MN       FE       SI       TI       AL
##        2        2        2        1        1        1        1        1
##       NI        V       PB
##        1        1        1
##
## Node number 1: 8647 observations,    complexity param=0.3648961
##   mean=4.549602, MSE=25.32681
##   left son=2 (8390 obs) right son=3 (257 obs)
##   Primary splits:
##       OC < 3.693035 to the left,  improve=0.3648961, (0 missing)
##       OP < 0.838775 to the left,  improve=0.3426816, (0 missing)
##       EC < 0.477345 to the left,  improve=0.3353746, (0 missing)
##       K  < 0.078975 to the left,  improve=0.3108213, (0 missing)
##       ZN < 0.003445 to the left,  improve=0.2197968, (0 missing)
##   Surrogate splits:
##       OP  < 0.815255 to the left,  agree=0.986, adj=0.545, (0 split)
##       EC  < 0.8572   to the left,  agree=0.978, adj=0.257, (0 split)
##       PB  < 0.03269  to the left,  agree=0.971, adj=0.027, (0 split)
##       K   < 0.70079  to the left,  agree=0.971, adj=0.027, (0 split)
##       NO3 < 10.94156 to the left,  agree=0.971, adj=0.027, (0 split)
##
## Node number 2: 8390 observations,    complexity param=0.1541707
##   mean=4.017542, MSE=10.74484
##   left son=4 (6378 obs) right son=5 (2012 obs)
##   Primary splits:
##       SO4 < 1.0019   to the left,  improve=0.3745293, (0 missing)
##       S   < 0.35026  to the left,  improve=0.3738907, (0 missing)
##       K   < 0.032105 to the left,  improve=0.3698196, (0 missing)
```

```
##       OC  < 1.06415  to the left,  improve=0.3622052, (0 missing)
##       OP  < 0.237865 to the left,  improve=0.3482584, (0 missing)
##   Surrogate splits:
##       S        < 0.379    to the left,  agree=0.979, adj=0.915, (0 split)
##       SiteCode splits as  LL-LLLLRRLLLLRLLLRRLR-LLLRLLLRLLRLLLL--LRLRRRRLLLLR-LL
LLLLLLRL--LRLLLLRLLL-LRLLLLLLRLLLLLRLLLLLLLLLLLRLLLLLLLLLL-LLLLRLLLLRLLLLLRLLLLLRLL
L-RLLRLRLLLLLLL-L-LR-LLRLLLLLLRLL-L, agree=0.816, adj=0.233, (0 split)
##       OP       < 0.27197  to the left,  agree=0.813, adj=0.221, (0 split)
##       SE       < 0.000395 to the left,  agree=0.807, adj=0.193, (0 split)
##       ZN       < 0.004075 to the left,  agree=0.800, adj=0.164, (0 split)
##
## Node number 3: 257 observations,    complexity param=0.1222049
##   mean=21.91918, MSE=190.4247
##   left son=6 (214 obs) right son=7 (43 obs)
##   Primary splits:
##       OP       < 2.8095   to the left,  improve=0.5468624, (0 missing)
##       OC       < 15.0474  to the left,  improve=0.5137808, (0 missing)
##       K        < 0.18656  to the left,  improve=0.3267747, (0 missing)
##       SiteCode splits as  L--LR---LLL-LLLLLLL-R--LLLLLLL---RLLR--LL--L--LLRR--R-
RLR-------LLLL-RLLRL-RLRLLRL-----RL-LLR---LL-LL-L-LLR-LLL-----L-L--LL-L-R----L-LLL
---LLRL-LL-LLLL--L-R--LLL--L--LLLL-L, improve=0.3126466, (0 missing)
##       CL       < 0.014375 to the left,  improve=0.2974700, (0 missing)
##   Surrogate splits:
##       OC       < 12.1005  to the left,  agree=0.988, adj=0.930, (0 split)
##       SiteCode splits as  L--LL---LLL-LLLLLLL-L--LLLLLLL---LLRL--LL--L--LLRL--L-
LLR-------LLLL-RLLRL-LLRLLLL-----LL-LLR---LL-LL-L-LLL-LLL-----L-L--LL-L-R----L-LLL
---LLRL-LL-LLRL--L-R--LLL--L--LLLL-L, agree=0.879, adj=0.279, (0 split)
##       EC       < 2.921035 to the left,  agree=0.844, adj=0.070, (0 split)
##       Date     splits as  L-LL--L-LLL-L-L---L-LL--LLLLLL-LLLLL--LLLLRL-LLLLLL--L
--LLLLL--L--------------LLL-L-LL-L-LL-LL-LL-LLLLLLLLLLLLRLLL-L-LLLL, agree=0.840, a
dj=0.047, (0 split)
##       NI       < -6.5e-05 to the right, agree=0.837, adj=0.023, (0 split)
##
## Node number 4: 6378 observations,    complexity param=0.04649899
##   mean=2.890826, MSE=4.189719
##   left son=8 (4237 obs) right son=9 (2141 obs)
##   Primary splits:
##       K  < 0.02811  to the left,  improve=0.3810835, (0 missing)
##       OC < 0.93938  to the left,  improve=0.3553606, (0 missing)
##       OP < 0.16584  to the left,  improve=0.3182269, (0 missing)
##       BR < 0.001165 to the left,  improve=0.2981336, (0 missing)
##       EC < 0.11325  to the left,  improve=0.2856111, (0 missing)
##   Surrogate splits:
##       MN < 0.000965 to the left,  agree=0.814, adj=0.446, (0 split)
##       FE < 0.037735 to the left,  agree=0.813, adj=0.444, (0 split)
##       SI < 0.14337  to the left,  agree=0.811, adj=0.437, (0 split)
##       TI < 0.003515 to the left,  agree=0.811, adj=0.437, (0 split)
##       AL < 0.058885 to the left,  agree=0.809, adj=0.430, (0 split)
##
## Node number 5: 2012 observations,    complexity param=0.03463939
##   mean=7.589211, MSE=14.74337
```

```
##   left son=10 (1728 obs) right son=11 (284 obs)
##   Primary splits:
##       SO4 < 2.38635  to the left,  improve=0.2557358, (0 missing)
##       CR  < 0.000665 to the left,  improve=0.2396845, (0 missing)
##       S   < 0.67805  to the left,  improve=0.2374953, (0 missing)
##       V   < 0.0029   to the left,  improve=0.2366923, (0 missing)
##       K   < 0.148315 to the left,  improve=0.2313183, (0 missing)
##   Surrogate splits:
##       S        < 0.874975 to the left,  agree=0.970, adj=0.785, (0 split)
##       SiteCode splits as  LL-LLLLLLLLL-LL--LLLR-LLLL--LLL-LLLL---LLLLLLLL-LLL-L-
L-LLL-LL--LLLLL-LLL--LL-LLL-LL-LLLLL-L-L--L-L--LLLLLLLLLLL-LLLLLLLLLL-LLLLLLL-LLL-L
L-LLLL-LL-LLLLL-L-LL-LLL-LLL-LL-L-L, agree=0.869, adj=0.074, (0 split)
##       NI       < 0.0017   to the left,  agree=0.869, adj=0.074, (0 split)
##       V        < 0.004775 to the left,  agree=0.869, adj=0.074, (0 split)
##       SE       < 0.002    to the left,  agree=0.867, adj=0.060, (0 split)
##
## Node number 6: 214 observations,    complexity param=0.02313744
##   mean=17.34485, MSE=60.01467
##   left son=12 (201 obs) right son=13 (13 obs)
##   Primary splits:
##       NO3      < 5.11497  to the left,  improve=0.3945392, (0 missing)
##       SiteCode splits as  L--LL---LLL-LLLLLLL-R--LLLLLLL---LLLL--LL--L--LLLR--L-
LL--------LLLL--LLRL-LL-LLRL-----RL-LLL---LL-LL-R-LLL-LLL-----L-L--LL-L------L-LLL
---LLLL-LL-LLLL--L----LLL--L--LLLL-L, improve=0.3915086, (0 missing)
##       Date     splits as  R-LL--L-LRL-L-L---L-LL--LLLLLL-RLRLL--LLRL-L-RLRLRL--L
--LLLLL--L--------------LLL-L-LL-L-LL-LL-LL-LLLLLLLLLLLL-LLL-L-LLLL, improve=0.3623
386, (0 missing)
##       K        < 0.25385  to the left,  improve=0.3110933, (0 missing)
##       SO4      < 4.978265 to the left,  improve=0.2807596, (0 missing)
##   Surrogate splits:
##       Date     splits as  L-LL--L-LRL-L-L---L-LL--LLLLLL-LLRLL--LLLL-L-LRRLLL--L
--LLLLL--L--------------LLL-L-LL-L-LL-LL-LL-LLLLLLLLLLLL-LLL-L-LLLL, agree=0.958, a
dj=0.308, (0 split)
##       SE       < 0.00251  to the left,  agree=0.958, adj=0.308, (0 split)
##       S        < 2.37981  to the left,  agree=0.958, adj=0.308, (0 split)
##       SO4      < 6.5974   to the left,  agree=0.953, adj=0.231, (0 split)
##       SiteCode splits as  L--LL---LLL-LLLLLLL-L--LLLLLLL---LLLL--LL--L--LLLR--L-
LL--------LLLL--LLLL-LL-LLLL-----LL-LLL---LL-LL-L-LLL-LLL-----L-L--LL-L------L-LLL
---LLLL-LL-LLLL--L----LLL--L--LLLL-L, agree=0.949, adj=0.154, (0 split)
##
## Node number 7: 43 observations,    complexity param=0.02850279
##   mean=44.68448, MSE=217.0472
##   left son=14 (33 obs) right son=15 (10 obs)
##   Primary splits:
##       OC       < 25.39485 to the left,  improve=0.6688224, (0 missing)
##       OP       < 7.920375 to the left,  improve=0.6323192, (0 missing)
##       EC       < 1.432395 to the left,  improve=0.5185635, (0 missing)
##       SiteCode splits as  ----R----L----------R--L---------L-LR---------L-L---R-
R-L-------R----L--R--L-L--L------L----L-------------R------------------R----L-L
-------L-------L-----L-------------L--, improve=0.4939978, (0 missing)
##       CL       < 0.02494  to the left,  improve=0.4923968, (0 missing)
```

```
##   Surrogate splits:
##       OP       < 6.893135 to the left,  agree=0.953, adj=0.8, (0 split)
##       SiteCode splits as  ----R----L----------L--L---------L-LR---------L-L---R-
L-L-------L----L--L--L-L--L------L----L-------------R------------------R----L-L
-------L-------L-----L-------------L--, agree=0.884, adj=0.5, (0 split)
##       EC       < 1.903125 to the left,  agree=0.884, adj=0.5, (0 split)
##       Date     splits as  ----------------------------------------L----------L
------------------------------R--L--L--L----LL---LLRLR--LL------LR-, agree=0.860, a
dj=0.4, (0 split)
##       CL       < 0.031235 to the left,  agree=0.837, adj=0.3, (0 split)
##
## Node number 8: 4237 observations,    complexity param=0.01140478
##   mean=1.992608, MSE=1.44871
##   left son=16 (2207 obs) right son=17 (2030 obs)
##   Primary splits:
##       OP  < 0.09625  to the left,  improve=0.4069047, (0 missing)
##       K   < 0.012955 to the left,  improve=0.4025839, (0 missing)
##       OC  < 0.532155 to the left,  improve=0.3954182, (0 missing)
##       S   < 0.098965 to the left,  improve=0.3856653, (0 missing)
##       SO4 < 0.26405  to the left,  improve=0.3640022, (0 missing)
##   Surrogate splits:
##       OC  < 0.423575 to the left,  agree=0.875, adj=0.739, (0 split)
##       EC  < 0.048955 to the left,  agree=0.793, adj=0.568, (0 split)
##       K   < 0.012705 to the left,  agree=0.745, adj=0.468, (0 split)
##       S   < 0.11291  to the left,  agree=0.734, adj=0.445, (0 split)
##       SO4 < 0.29305  to the left,  agree=0.726, adj=0.429, (0 split)
##
## Node number 9: 2141 observations,    complexity param=0.01306348
##   mean=4.668382, MSE=4.857784
##   left son=18 (1819 obs) right son=19 (322 obs)
##   Primary splits:
##       OC < 1.81129  to the left,  improve=0.2750742, (0 missing)
##       OP < 0.28626  to the left,  improve=0.1937325, (0 missing)
##       FE < 0.20788  to the left,  improve=0.1722669, (0 missing)
##       K  < 0.06402  to the left,  improve=0.1667441, (0 missing)
##       EC < 0.22105  to the left,  improve=0.1650228, (0 missing)
##   Surrogate splits:
##       OP   < 0.38288  to the left,  agree=0.921, adj=0.475, (0 split)
##       EC   < 0.31343  to the left,  agree=0.880, adj=0.202, (0 split)
##       Date splits as  LLLLLRLLLLLLLLLLLLLLLLLLLLLLLLLLLLL--LLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLRLLRRLLLLLLLLLLLLLL, agree=0.853, adj=
0.022, (0 split)
##       CR   < 0.000805 to the left,  agree=0.852, adj=0.016, (0 split)
##       SE   < 0.00147  to the left,  agree=0.851, adj=0.012, (0 split)
##
## Node number 10: 1728 observations,    complexity param=0.01685673
##   mean=6.802017, MSE=8.031354
##   left son=20 (964 obs) right son=21 (764 obs)
##   Primary splits:
##       OC < 1.16163  to the left,  improve=0.2660032, (0 missing)
##       K  < 0.0549   to the left,  improve=0.2628756, (0 missing)
```

```
##       FE < 0.404225 to the left,  improve=0.2057942, (0 missing)
##       SI < 1.209615 to the left,  improve=0.2053201, (0 missing)
##       AL < 0.636465 to the left,  improve=0.2052434, (0 missing)
##   Surrogate splits:
##       OP       < 0.32231  to the left,  agree=0.850, adj=0.661, (0 split)
##       EC       < 0.206845 to the left,  agree=0.789, adj=0.522, (0 split)
##       SiteCode splits as  RL-LLLLLRLLL-LL--RLLR-RLLL--LLL-RRLL---RLLLRLLL-LRL-L-
L-LLL-RL--LLLLL-RLL--LL-RRL-LR-LLLLR-R-L--L-L--RLLLLLLRLLL-RLLRLRLLLL-LLLLLLL-LRL-R
L-RLLR-LL-RLLLL-L-LR-LLR-LLL-LL-R-L, agree=0.683, adj=0.284, (0 split)
##       CU       < 0.000695 to the left,  agree=0.676, adj=0.267, (0 split)
##       ZN       < 0.00406  to the left,  agree=0.674, adj=0.263, (0 split)
##
## Node number 11: 284 observations,    complexity param=0.0173121
##   mean=12.37889, MSE=28.8712
##   left son=22 (270 obs) right son=23 (14 obs)
##   Primary splits:
##       SO4 < 6.76665  to the left,  improve=0.4623944, (0 missing)
##       PB  < 0.016615 to the left,  improve=0.4265695, (0 missing)
##       MN  < 0.008395 to the left,  improve=0.4185620, (0 missing)
##       ZN  < 0.04141  to the left,  improve=0.4171126, (0 missing)
##       S   < 1.93679  to the left,  improve=0.4099518, (0 missing)
##   Surrogate splits:
##       S    < 2.573415 to the left,  agree=0.993, adj=0.857, (0 split)
##       NI   < 0.00304  to the left,  agree=0.975, adj=0.500, (0 split)
##       V    < 0.008205 to the left,  agree=0.975, adj=0.500, (0 split)
##       ZN   < 0.053945 to the left,  agree=0.975, adj=0.500, (0 split)
##       Date splits as  LL-LLLLL--L-R-----L-L---------L-------LLL-L--LL-LLLL-LLLL-
LLL-LL-LLLLLLLL-LLLRLLLLRLLLLRLLLLLLLLLLLLLLLLLLRLLLLLLRL-LLLLLL, agree=0.972, adj=
0.429, (0 split)
##
## Node number 12: 201 observations
##   mean=16.10734, MSE=28.57004
##
## Node number 13: 13 observations
##   mean=36.4786, MSE=156.4184
##
## Node number 14: 33 observations
##   mean=38.052, MSE=73.78158
##
## Node number 15: 10 observations
##   mean=66.57164, MSE=65.60985
##
## Node number 16: 2207 observations
##   mean=1.256258, MSE=0.627724
##
## Node number 17: 2030 observations
##   mean=2.793161, MSE=1.110907
##
## Node number 18: 1819 observations
##   mean=4.182025, MSE=3.344767
##
```

```
## Node number 19: 322 observations
##   mean=7.415849, MSE=4.520096
##
## Node number 20: 964 observations
##   mean=5.500812, MSE=4.404893
##
## Node number 21: 764 observations
##   mean=8.443853, MSE=7.775161
##
## Node number 22: 270 observations
##   mean=11.5469, MSE=12.47624
##
## Node number 23: 14 observations
##   mean=28.42452, MSE=74.24786
```

```
pred <- predict(fit1, US_DATA_LRG_test)

ModelMetrics::rmse(pred,US_DATA_LRG_test$PM2.5_UNC)
```

```
## [1] 6.17434
```

```
ModelMetrics::gini(pred,US_DATA_LRG_test$PM2.5_UNC)
```

```
## [1] 0.9282768
```

```
#0.03234565
fit1$cptable
```

```
##            CP nsplit rel error    xerror       xstd
## 1  0.36489610      0 1.0000000 1.0003518 0.07605944
## 2  0.15417066      1 0.6351039 0.6501399 0.04173009
## 3  0.12220485      2 0.4809332 0.4985611 0.03998255
## 4  0.04649899      3 0.3587284 0.3941103 0.02711683
## 5  0.03463939      4 0.3122294 0.3421852 0.02628566
## 6  0.02850279      5 0.2775900 0.3213551 0.02532643
## 7  0.02313744      6 0.2490872 0.2992325 0.02255095
## 8  0.01731210      7 0.2259498 0.2773869 0.02106606
## 9  0.01685673      8 0.2086377 0.2520433 0.01790144
## 10 0.01306348      9 0.1917809 0.2305255 0.01659266
## 11 0.01140478     10 0.1787175 0.2197864 0.01640755
## 12 0.01000000     11 0.1673127 0.2171949 0.01646286
```

```
 fit2 <- rpart(
    formula = PM2.5 ~ .-PM2.5_UNC,
    data    = US_DATA_LRG,
    method  = "anova",
    control = list(minsplit = 10, maxdepth = 12, xval = 10)
)
fit2
```

```
## n= 8647
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 8647 219000.9000  4.549602
##    2) OC< 3.693035 8390  90149.2100  4.017542
##      4) SO4< 1.0019 6378  26722.0200  2.890826
##        8) K< 0.02811 4237   6138.1860  1.992608
##         16) OP< 0.09625 2207   1385.3870  1.256258 *
##         17) OP>=0.09625 2030   2255.1420  2.793161 *
##        9) K>=0.02811 2141  10400.5200  4.668382
##         18) OC< 1.81129 1819   6084.1310  4.182025 *
##         19) OC>=1.81129 322   1455.4710  7.415849 *
##      5) SO4>=1.0019 2012  29663.6600  7.589211
##       10) SO4< 2.38635 1728  13878.1800  6.802017
##         20) OC< 1.16163 964   4246.3170  5.500812 *
##         21) OC>=1.16163 764   5940.2230  8.443853 *
##       11) SO4>=2.38635 284   8199.4220 12.378890
##         22) SO4< 6.76665 270   3368.5850 11.546900 *
##         23) SO4>=6.76665 14   1039.4700 28.424520 *
##    3) OC>=3.693035 257  48939.1500 21.919180
##      6) OP< 2.8095 214  12843.1400 17.344850
##       12) NO3< 5.11497 201   5742.5780 16.107340 *
##       13) NO3>=5.11497 13   2033.4390 36.478600 *
##      7) OP>=2.8095 43   9333.0300 44.684480
##       14) OC< 25.39485 33   2434.7920 38.052000 *
##       15) OC>=25.39485 10    656.0985 66.571640 *
```

```
fit2$cptable
```

```
##            CP nsplit rel error    xerror       xstd
## 1  0.36489610      0 1.0000000 1.0001045 0.07604518
## 2  0.15417066      1 0.6351039 0.6436667 0.04050769
## 3  0.12220485      2 0.4809332 0.4934282 0.03891217
## 4  0.04649899      3 0.3587284 0.3756975 0.02425669
## 5  0.03463939      4 0.3122294 0.3228140 0.02394118
## 6  0.02850279      5 0.2775900 0.2981885 0.02293460
## 7  0.02313744      6 0.2490872 0.2716021 0.02061702
## 8  0.01731210      7 0.2259498 0.2524880 0.01702999
## 9  0.01685673      8 0.2086377 0.2389885 0.01607148
## 10 0.01306348      9 0.1917809 0.2217493 0.01557486
## 11 0.01140478     10 0.1787175 0.2090751 0.01550154
## 12 0.01000000     11 0.1673127 0.2050727 0.01531889
```

2. use a grid search method to find the optimal hyper-parameters for a single tree model

```
hyper_grid <- expand.grid(
  minsplit = seq(5, 20, 1),
  maxdepth = seq(8, 15, 1)
)

head(hyper_grid)
```

```
##   minsplit maxdepth
## 1        5        8
## 2        6        8
## 3        7        8
## 4        8        8
## 5        9        8
## 6       10        8
```

```
# total number of combinations
nrow(hyper_grid)
```

```
## [1] 128
```

```
models <- list() #best method i've found for doing this--but computationally expens
ive...

for (i in 1:nrow(hyper_grid)) {

  # get minsplit, maxdepth values at row i
  minsplit <- hyper_grid$minsplit[i]
  maxdepth <- hyper_grid$maxdepth[i]

  # train a model and store in the list
  models[[i]] <- rpart(
   formula = PM2.5 ~ .-PM2.5_UNC,
    data    = US_DATA_LRG,
    method  = "anova",
    control = list(minsplit = minsplit, maxdepth = maxdepth)
    )
}
```

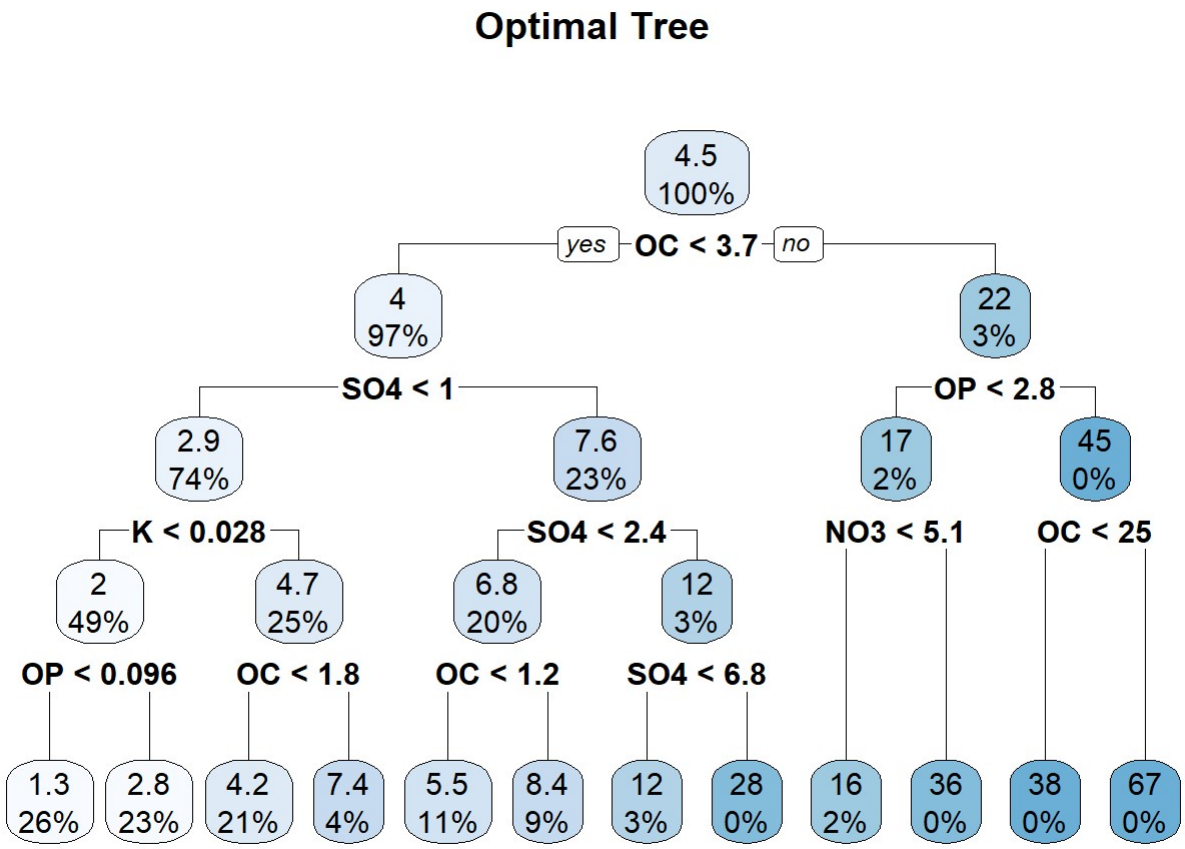
```
# function to get optimal cp
get_cp <- function(x) {
  min    <- which.min(x$cptable[, "xerror"])
  cp <- x$cptable[min, "CP"]
}

# function to get minimum error
get_min_error <- function(x) {
  min    <- which.min(x$cptable[, "xerror"])
  xerror <- x$cptable[min, "xerror"]
}

hyper_grid %>%
  mutate(
    cp    = purrr::map_dbl(models, get_cp),
    error = purrr::map_dbl(models, get_min_error)
    ) %>%
  arrange(error) %>%
  top_n(-5, wt = error)
```

```
##   minsplit maxdepth   cp     error
## 1       10        8 0.01 0.1973481
## 2       14       15 0.01 0.2001115
## 3       14       14 0.01 0.2009031
## 4        7       12 0.01 0.2020608
## 5       20       13 0.01 0.2027092
```

```
optimal_tree <- rpart(
    formula = PM2.5 ~ .-PM2.5_UNC,
    data    = US_DATA_LRG,
    method  = "anova",
    control = list(minsplit = 10, maxdepth = 8, cp = 0.01, xval=10)
    )

pred <- predict(optimal_tree, newdata = US_DATA_LRG_test)


rmse_op=RMSE(pred = pred, obs = US_DATA_LRG_test$PM2.5)
ModelMetrics::gini(pred,US_DATA_LRG_test$PM2.5)
```

```
## [1] 0.9329352
```

```
mae_op=MAE(pred = pred, obs = US_DATA_LRG_test$PM2.5)

rpart.plot(optimal_tree, main='Optimal Tree') #optimal tree determined througt grid search
```




```
summary(optimal_tree)
```

```
## Call:
## rpart(formula = PM2.5 ~ . - PM2.5_UNC, data = US_DATA_LRG, method = "anova",
##     control = list(minsplit = 10, maxdepth = 8, cp = 0.01, xval = 10))
##   n= 8647
##
##            CP nsplit rel error    xerror       xstd
## 1  0.36489610      0 1.0000000 1.0003115 0.07605303
## 2  0.15417066      1 0.6351039 0.6633240 0.04142286
## 3  0.12220485      2 0.4809332 0.5096679 0.03960257
## 4  0.04649899      3 0.3587284 0.3915864 0.02526373
## 5  0.03463939      4 0.3122294 0.3327126 0.02471715
## 6  0.02850279      5 0.2775900 0.3103268 0.02371196
## 7  0.02313744      6 0.2490872 0.2873444 0.02177956
## 8  0.01731210      7 0.2259498 0.2581629 0.01895770
## 9  0.01685673      8 0.2086377 0.2392860 0.01490210
## 10 0.01306348      9 0.1917809 0.2257883 0.01409075
## 11 0.01140478     10 0.1787175 0.2145467 0.01502649
## 12 0.01000000     11 0.1673127 0.2063296 0.01453057
##
## Variable importance
##       OC       OP      SO4        S       EC SiteCode        K       SE
##       28       21       11       10        7        5        3        2
##       ZN      NO3     Date       MN       FE       SI       TI       AL
##        2        2        2        1        1        1        1        1
##       NI        V       PB
##        1        1        1
##
## Node number 1: 8647 observations,    complexity param=0.3648961
##   mean=4.549602, MSE=25.32681
##   left son=2 (8390 obs) right son=3 (257 obs)
##   Primary splits:
##       OC < 3.693035 to the left,  improve=0.3648961, (0 missing)
##       OP < 0.838775 to the left,  improve=0.3426816, (0 missing)
##       EC < 0.477345 to the left,  improve=0.3353746, (0 missing)
##       K  < 0.078975 to the left,  improve=0.3108213, (0 missing)
##       ZN < 0.003445 to the left,  improve=0.2197968, (0 missing)
##   Surrogate splits:
##       OP  < 0.815255 to the left,  agree=0.986, adj=0.545, (0 split)
##       EC  < 0.8572   to the left,  agree=0.978, adj=0.257, (0 split)
##       PB  < 0.03269  to the left,  agree=0.971, adj=0.027, (0 split)
##       K   < 0.70079  to the left,  agree=0.971, adj=0.027, (0 split)
##       NO3 < 10.94156 to the left,  agree=0.971, adj=0.027, (0 split)
##
## Node number 2: 8390 observations,    complexity param=0.1541707
##   mean=4.017542, MSE=10.74484
##   left son=4 (6378 obs) right son=5 (2012 obs)
##   Primary splits:
##       SO4 < 1.0019   to the left,  improve=0.3745293, (0 missing)
##       S   < 0.35026  to the left,  improve=0.3738907, (0 missing)
##       K   < 0.032105 to the left,  improve=0.3698196, (0 missing)
```

```
##       OC  < 1.06415  to the left,  improve=0.3622052, (0 missing)
##       OP  < 0.237865 to the left,  improve=0.3482584, (0 missing)
##   Surrogate splits:
##       S        < 0.379    to the left,  agree=0.979, adj=0.915, (0 split)
##       SiteCode splits as  LL-LLLLRRLLLLRLLLRRLR-LLLRLLLRLLRLLLL--LRLRRRRLLLLR-LL
LLLLLLRL--LRLLLLRLLL-LRLLLLLLRLLLLLRLLLLLLLLLLLRLLLLLLLLLL-LLLLRLLLLRLLLLLRLLLLLRLL
L-RLLRLRLLLLLLL-L-LR-LLRLLLLLLRLL-L, agree=0.816, adj=0.233, (0 split)
##       OP       < 0.27197  to the left,  agree=0.813, adj=0.221, (0 split)
##       SE       < 0.000395 to the left,  agree=0.807, adj=0.193, (0 split)
##       ZN       < 0.004075 to the left,  agree=0.800, adj=0.164, (0 split)
##
## Node number 3: 257 observations,    complexity param=0.1222049
##   mean=21.91918, MSE=190.4247
##   left son=6 (214 obs) right son=7 (43 obs)
##   Primary splits:
##       OP       < 2.8095   to the left,  improve=0.5468624, (0 missing)
##       OC       < 15.0474  to the left,  improve=0.5137808, (0 missing)
##       K        < 0.18656  to the left,  improve=0.3267747, (0 missing)
##       SiteCode splits as  L--LR---LLL-LLLLLLL-R--LLLLLLL---RLLR--LL--L--LLRR--R-
RLR-------LLLL-RLLRL-RLRLLRL-----RL-LLR---LL-LL-L-LLR-LLL-----L-L--LL-L-R----L-LLL
---LLRL-LL-LLLL--L-R--LLL--L--LLLL-L, improve=0.3126466, (0 missing)
##       CL       < 0.014375 to the left,  improve=0.2974700, (0 missing)
##   Surrogate splits:
##       OC       < 12.1005  to the left,  agree=0.988, adj=0.930, (0 split)
##       SiteCode splits as  L--LL---LLL-LLLLLLL-L--LLLLLLL---LLRL--LL--L--LLRL--L-
LLR-------LLLL-RLLRL-LLRLLLL-----LL-LLR---LL-LL-L-LLL-LLL-----L-L--LL-L-R----L-LLL
---LLRL-LL-LLRL--L-R--LLL--L--LLLL-L, agree=0.879, adj=0.279, (0 split)
##       EC       < 2.921035 to the left,  agree=0.844, adj=0.070, (0 split)
##       Date     splits as  L-LL--L-LLL-L-L---L-LL--LLLLLL-LLLLL--LLLLRL-LLLLLL--L
--LLLLL--L--------------LLL-L-LL-L-LL-LL-LL-LLLLLLLLLLLLRLLL-L-LLLL, agree=0.840, a
dj=0.047, (0 split)
##       NI       < -6.5e-05 to the right, agree=0.837, adj=0.023, (0 split)
##
## Node number 4: 6378 observations,    complexity param=0.04649899
##   mean=2.890826, MSE=4.189719
##   left son=8 (4237 obs) right son=9 (2141 obs)
##   Primary splits:
##       K  < 0.02811  to the left,  improve=0.3810835, (0 missing)
##       OC < 0.93938  to the left,  improve=0.3553606, (0 missing)
##       OP < 0.16584  to the left,  improve=0.3182269, (0 missing)
##       BR < 0.001165 to the left,  improve=0.2981336, (0 missing)
##       EC < 0.11325  to the left,  improve=0.2856111, (0 missing)
##   Surrogate splits:
##       MN < 0.000965 to the left,  agree=0.814, adj=0.446, (0 split)
##       FE < 0.037735 to the left,  agree=0.813, adj=0.444, (0 split)
##       SI < 0.14337  to the left,  agree=0.811, adj=0.437, (0 split)
##       TI < 0.003515 to the left,  agree=0.811, adj=0.437, (0 split)
##       AL < 0.058885 to the left,  agree=0.809, adj=0.430, (0 split)
##
## Node number 5: 2012 observations,    complexity param=0.03463939
##   mean=7.589211, MSE=14.74337
```

```
##   left son=10 (1728 obs) right son=11 (284 obs)
##   Primary splits:
##       SO4 < 2.38635  to the left,  improve=0.2557358, (0 missing)
##       CR  < 0.000665 to the left,  improve=0.2396845, (0 missing)
##       S   < 0.67805  to the left,  improve=0.2374953, (0 missing)
##       V   < 0.0029   to the left,  improve=0.2366923, (0 missing)
##       K   < 0.148315 to the left,  improve=0.2313183, (0 missing)
##   Surrogate splits:
##       S        < 0.874975 to the left,  agree=0.970, adj=0.785, (0 split)
##       SiteCode splits as  LL-LLLLLLLLL-LL--LLLR-LLLL--LLL-LLLL---LLLLLLLL-LLL-L-
L-LLL-LL--LLLLL-LLL--LL-LLL-LL-LLLLL-L-L--L-L--LLLLLLLLLLL-LLLLLLLLLL-LLLLLLL-LLL-L
L-LLLL-LL-LLLLL-L-LL-LLL-LLL-LL-L-L, agree=0.869, adj=0.074, (0 split)
##       NI       < 0.0017   to the left,  agree=0.869, adj=0.074, (0 split)
##       V        < 0.004775 to the left,  agree=0.869, adj=0.074, (0 split)
##       SE       < 0.002    to the left,  agree=0.867, adj=0.060, (0 split)
##
## Node number 6: 214 observations,    complexity param=0.02313744
##   mean=17.34485, MSE=60.01467
##   left son=12 (201 obs) right son=13 (13 obs)
##   Primary splits:
##       NO3      < 5.11497  to the left,  improve=0.3945392, (0 missing)
##       SiteCode splits as  L--LL---LLL-LLLLLLL-R--LLLLLLL---LLLL--LL--L--LLLR--L-
LL--------LLLL--LLRL-LL-LLRL-----RL-LLL---LL-LL-R-LLL-LLL-----L-L--LL-L------L-LLL
---LLLL-LL-LLLL--L----LLL--L--LLLL-L, improve=0.3915086, (0 missing)
##       Date     splits as  R-LL--L-LRL-L-L---L-LL--LLLLLL-RLRLL--LLRL-L-RLRLRL--L
--LLLLL--L--------------LLL-L-LL-L-LL-LL-LL-LLLLLLLLLLLL-LLL-L-LLLL, improve=0.3623
386, (0 missing)
##       K        < 0.25385  to the left,  improve=0.3110933, (0 missing)
##       SO4      < 4.978265 to the left,  improve=0.2807596, (0 missing)
##   Surrogate splits:
##       Date     splits as  L-LL--L-LRL-L-L---L-LL--LLLLLL-LLRLL--LLLL-L-LRRLLL--L
--LLLLL--L--------------LLL-L-LL-L-LL-LL-LL-LLLLLLLLLLLL-LLL-L-LLLL, agree=0.958, a
dj=0.308, (0 split)
##       SE       < 0.00251  to the left,  agree=0.958, adj=0.308, (0 split)
##       S        < 2.37981  to the left,  agree=0.958, adj=0.308, (0 split)
##       SO4      < 6.5974   to the left,  agree=0.953, adj=0.231, (0 split)
##       SiteCode splits as  L--LL---LLL-LLLLLLL-L--LLLLLLL---LLLL--LL--L--LLLR--L-
LL--------LLLL--LLLL-LL-LLLL-----LL-LLL---LL-LL-L-LLL-LLL-----L-L--LL-L------L-LLL
---LLLL-LL-LLLL--L----LLL--L--LLLL-L, agree=0.949, adj=0.154, (0 split)
##
## Node number 7: 43 observations,    complexity param=0.02850279
##   mean=44.68448, MSE=217.0472
##   left son=14 (33 obs) right son=15 (10 obs)
##   Primary splits:
##       OC       < 25.39485 to the left,  improve=0.6688224, (0 missing)
##       OP       < 7.920375 to the left,  improve=0.6323192, (0 missing)
##       EC       < 1.432395 to the left,  improve=0.5185635, (0 missing)
##       SiteCode splits as  ----R----L----------R--L---------L-LR---------L-L---R-
R-L-------R----L--R--L-L--L------L----L-------------R------------------R----L-L
-------L-------L-----L-------------L--, improve=0.4939978, (0 missing)
##       CL       < 0.02494  to the left,  improve=0.4923968, (0 missing)
```

```
##   Surrogate splits:
##       OP       < 6.893135 to the left,  agree=0.953, adj=0.8, (0 split)
##       SiteCode splits as  ----R----L----------L--L---------L-LR---------L-L---R-
L-L-------L----L--L--L-L--L------L----L-------------R------------------R----L-L
-------L-------L-----L-------------L--, agree=0.884, adj=0.5, (0 split)
##       EC       < 1.903125 to the left,  agree=0.884, adj=0.5, (0 split)
##       Date     splits as  ----------------------------------------L----------L
------------------------------R--L--L--L----LL---LLRLR--LL------LR-, agree=0.860, a
dj=0.4, (0 split)
##       CL       < 0.031235 to the left,  agree=0.837, adj=0.3, (0 split)
##
## Node number 8: 4237 observations,    complexity param=0.01140478
##   mean=1.992608, MSE=1.44871
##   left son=16 (2207 obs) right son=17 (2030 obs)
##   Primary splits:
##       OP  < 0.09625  to the left,  improve=0.4069047, (0 missing)
##       K   < 0.012955 to the left,  improve=0.4025839, (0 missing)
##       OC  < 0.532155 to the left,  improve=0.3954182, (0 missing)
##       S   < 0.098965 to the left,  improve=0.3856653, (0 missing)
##       SO4 < 0.26405  to the left,  improve=0.3640022, (0 missing)
##   Surrogate splits:
##       OC  < 0.423575 to the left,  agree=0.875, adj=0.739, (0 split)
##       EC  < 0.048955 to the left,  agree=0.793, adj=0.568, (0 split)
##       K   < 0.012705 to the left,  agree=0.745, adj=0.468, (0 split)
##       S   < 0.11291  to the left,  agree=0.734, adj=0.445, (0 split)
##       SO4 < 0.29305  to the left,  agree=0.726, adj=0.429, (0 split)
##
## Node number 9: 2141 observations,    complexity param=0.01306348
##   mean=4.668382, MSE=4.857784
##   left son=18 (1819 obs) right son=19 (322 obs)
##   Primary splits:
##       OC < 1.81129  to the left,  improve=0.2750742, (0 missing)
##       OP < 0.28626  to the left,  improve=0.1937325, (0 missing)
##       FE < 0.20788  to the left,  improve=0.1722669, (0 missing)
##       K  < 0.06402  to the left,  improve=0.1667441, (0 missing)
##       EC < 0.22105  to the left,  improve=0.1650228, (0 missing)
##   Surrogate splits:
##       OP   < 0.38288  to the left,  agree=0.921, adj=0.475, (0 split)
##       EC   < 0.31343  to the left,  agree=0.880, adj=0.202, (0 split)
##       Date splits as  LLLLLRLLLLLLLLLLLLLLLLLLLLLLLLLLLLL--LLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLRLLRRLLLLLLLLLLLLLL, agree=0.853, adj=
0.022, (0 split)
##       CR   < 0.000805 to the left,  agree=0.852, adj=0.016, (0 split)
##       SE   < 0.00147  to the left,  agree=0.851, adj=0.012, (0 split)
##
## Node number 10: 1728 observations,    complexity param=0.01685673
##   mean=6.802017, MSE=8.031354
##   left son=20 (964 obs) right son=21 (764 obs)
##   Primary splits:
##       OC < 1.16163  to the left,  improve=0.2660032, (0 missing)
##       K  < 0.0549   to the left,  improve=0.2628756, (0 missing)
```

```
##       FE < 0.404225 to the left,  improve=0.2057942, (0 missing)
##       SI < 1.209615 to the left,  improve=0.2053201, (0 missing)
##       AL < 0.636465 to the left,  improve=0.2052434, (0 missing)
##   Surrogate splits:
##       OP       < 0.32231  to the left,  agree=0.850, adj=0.661, (0 split)
##       EC       < 0.206845 to the left,  agree=0.789, adj=0.522, (0 split)
##       SiteCode splits as  RL-LLLLLRLLL-LL--RLLR-RLLL--LLL-RRLL---RLLLRLLL-LRL-L-
L-LLL-RL--LLLLL-RLL--LL-RRL-LR-LLLLR-R-L--L-L--RLLLLLLRLLL-RLLRLRLLLL-LLLLLLL-LRL-R
L-RLLR-LL-RLLLL-L-LR-LLR-LLL-LL-R-L, agree=0.683, adj=0.284, (0 split)
##       CU       < 0.000695 to the left,  agree=0.676, adj=0.267, (0 split)
##       ZN       < 0.00406  to the left,  agree=0.674, adj=0.263, (0 split)
##
## Node number 11: 284 observations,    complexity param=0.0173121
##   mean=12.37889, MSE=28.8712
##   left son=22 (270 obs) right son=23 (14 obs)
##   Primary splits:
##       SO4 < 6.76665  to the left,  improve=0.4623944, (0 missing)
##       PB  < 0.016615 to the left,  improve=0.4265695, (0 missing)
##       MN  < 0.008395 to the left,  improve=0.4185620, (0 missing)
##       ZN  < 0.04141  to the left,  improve=0.4171126, (0 missing)
##       S   < 1.93679  to the left,  improve=0.4099518, (0 missing)
##   Surrogate splits:
##       S    < 2.573415 to the left,  agree=0.993, adj=0.857, (0 split)
##       NI   < 0.00304  to the left,  agree=0.975, adj=0.500, (0 split)
##       V    < 0.008205 to the left,  agree=0.975, adj=0.500, (0 split)
##       ZN   < 0.053945 to the left,  agree=0.975, adj=0.500, (0 split)
##       Date splits as  LL-LLLLL--L-R-----L-L---------L-------LLL-L--LL-LLLL-LLLL-
LLL-LL-LLLLLLLL-LLLRLLLLRLLLLRLLLLLLLLLLLLLLLLLLRLLLLLLRL-LLLLLL, agree=0.972, adj=
0.429, (0 split)
##
## Node number 12: 201 observations
##   mean=16.10734, MSE=28.57004
##
## Node number 13: 13 observations
##   mean=36.4786, MSE=156.4184
##
## Node number 14: 33 observations
##   mean=38.052, MSE=73.78158
##
## Node number 15: 10 observations
##   mean=66.57164, MSE=65.60985
##
## Node number 16: 2207 observations
##   mean=1.256258, MSE=0.627724
##
## Node number 17: 2030 observations
##   mean=2.793161, MSE=1.110907
##
## Node number 18: 1819 observations
##   mean=4.182025, MSE=3.344767
##
```

```
## Node number 19: 322 observations
##   mean=7.415849, MSE=4.520096
##
## Node number 20: 964 observations
##   mean=5.500812, MSE=4.404893
##
## Node number 21: 764 observations
##   mean=8.443853, MSE=7.775161
##
## Node number 22: 270 observations
##   mean=11.5469, MSE=12.47624
##
## Node number 23: 14 observations
##   mean=28.42452, MSE=74.24786
```

```
optimal_tree
```

```
## n= 8647
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 8647 219000.9000  4.549602
##    2) OC< 3.693035 8390  90149.2100  4.017542
##      4) SO4< 1.0019 6378  26722.0200  2.890826
##        8) K< 0.02811 4237   6138.1860  1.992608
##         16) OP< 0.09625 2207   1385.3870  1.256258 *
##         17) OP>=0.09625 2030   2255.1420  2.793161 *
##        9) K>=0.02811 2141  10400.5200  4.668382
##         18) OC< 1.81129 1819   6084.1310  4.182025 *
##         19) OC>=1.81129 322   1455.4710  7.415849 *
##      5) SO4>=1.0019 2012  29663.6600  7.589211
##       10) SO4< 2.38635 1728  13878.1800  6.802017
##         20) OC< 1.16163 964   4246.3170  5.500812 *
##         21) OC>=1.16163 764   5940.2230  8.443853 *
##       11) SO4>=2.38635 284   8199.4220 12.378890
##         22) SO4< 6.76665 270   3368.5850 11.546900 *
##         23) SO4>=6.76665 14   1039.4700 28.424520 *
##    3) OC>=3.693035 257  48939.1500 21.919180
##      6) OP< 2.8095 214  12843.1400 17.344850
##       12) NO3< 5.11497 201   5742.5780 16.107340 *
##       13) NO3>=5.11497 13   2033.4390 36.478600 *
##      7) OP>=2.8095 43   9333.0300 44.684480
##       14) OC< 25.39485 33   2434.7920 38.052000 *
##       15) OC>=25.39485 10    656.0985 66.571640 *
```

```
tmp <- printcp(optimal_tree)
```

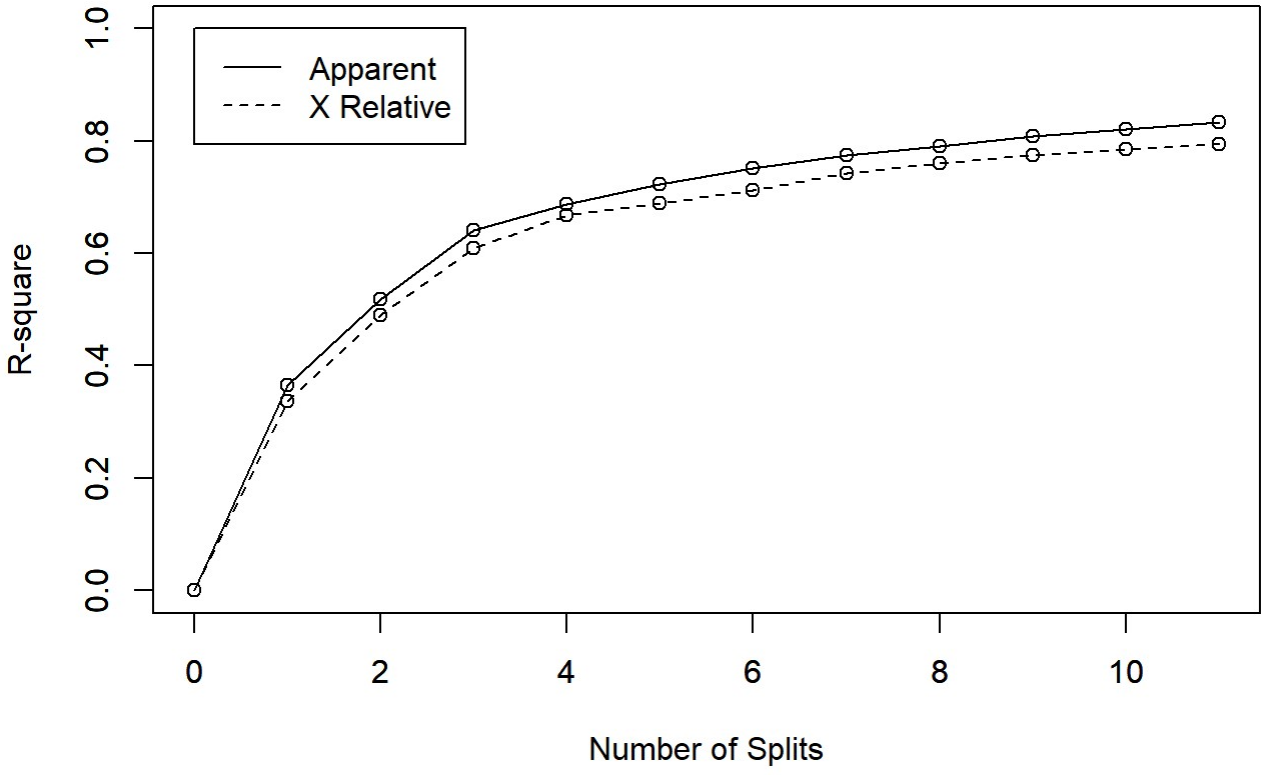
```
##
## Regression tree:
## rpart(formula = PM2.5 ~ . - PM2.5_UNC, data = US_DATA_LRG, method = "anova",
##     control = list(minsplit = 10, maxdepth = 8, cp = 0.01, xval = 10))
##
## Variables actually used in tree construction:
## [1] K   NO3 OC  OP  SO4
##
## Root node error: 219001/8647 = 25.327
##
## n= 8647
##
##          CP nsplit rel error  xerror     xstd
## 1  0.364896      0   1.00000 1.00031 0.076053
## 2  0.154171      1   0.63510 0.66332 0.041423
## 3  0.122205      2   0.48093 0.50967 0.039603
## 4  0.046499      3   0.35873 0.39159 0.025264
## 5  0.034639      4   0.31223 0.33271 0.024717
## 6  0.028503      5   0.27759 0.31033 0.023712
## 7  0.023137      6   0.24909 0.28734 0.021780
## 8  0.017312      7   0.22595 0.25816 0.018958
## 9  0.016857      8   0.20864 0.23929 0.014902
## 10 0.013063      9   0.19178 0.22579 0.014091
## 11 0.011405     10   0.17872 0.21455 0.015026
## 12 0.010000     11   0.16731 0.20633 0.014531
```

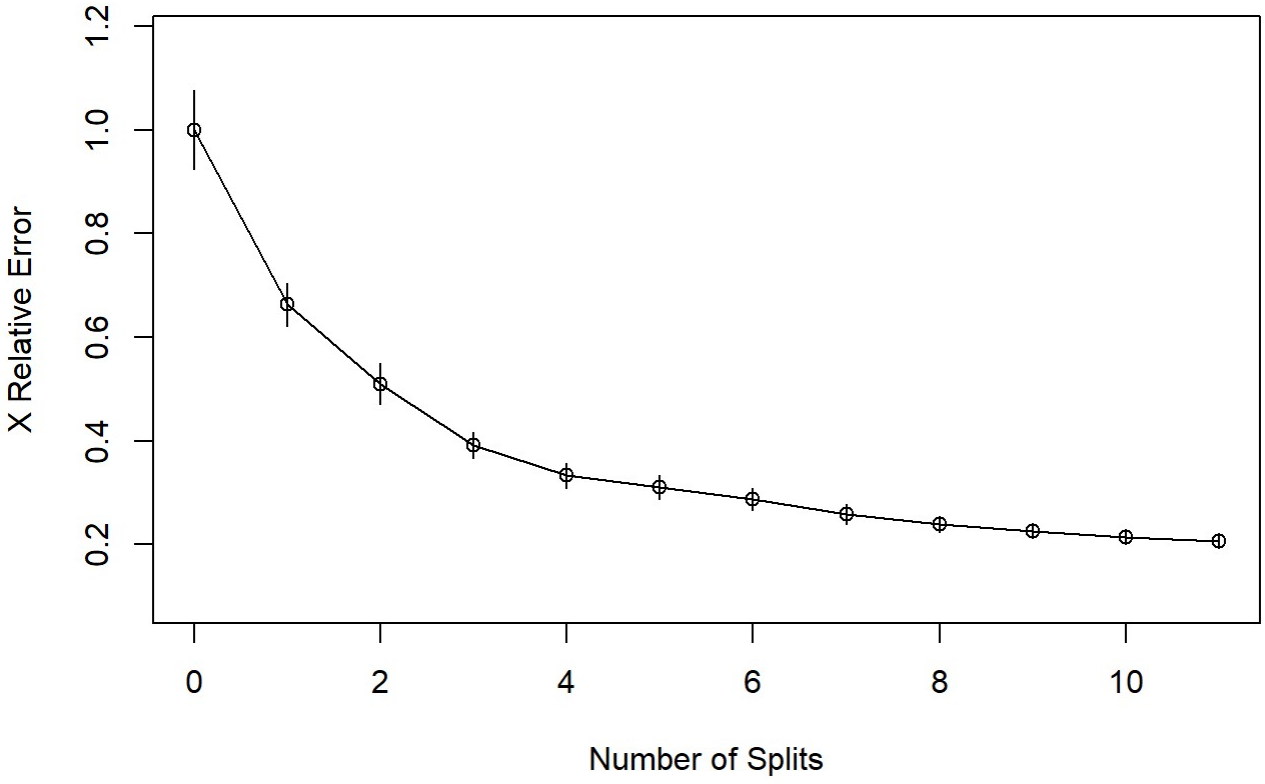
```
rsq.val <- 1-tmp[,c(3,4)]
rsq.val #rquared and xerror for each split
```

```
##    rel error        xerror
## 1  0.0000000 -0.0003114605
## 2  0.3648961  0.3366760066
## 3  0.5190668  0.4903321091
## 4  0.6412716  0.6084136421
## 5  0.6877706  0.6672873817
## 6  0.7224100  0.6896732000
## 7  0.7509128  0.7126555606
## 8  0.7740502  0.7418370628
## 9  0.7913623  0.7607139969
## 10 0.8082191  0.7742116659
## 11 0.8212825  0.7854533238
## 12 0.8326873  0.7936703885
```

```
rsq_op = rsq.val[nrow(rsq.val),] #final rquared and xerror
rsq.rpart(optimal_tree)#xerror and rsqu vs splits plot
```

```
##
## Regression tree:
## rpart(formula = PM2.5 ~ . - PM2.5_UNC, data = US_DATA_LRG, method = "anova",
##     control = list(minsplit = 10, maxdepth = 8, cp = 0.01, xval = 10))
##
## Variables actually used in tree construction:
## [1] K   NO3 OC  OP  SO4
##
## Root node error: 219001/8647 = 25.327
##
## n= 8647
##
##          CP nsplit rel error  xerror     xstd
## 1  0.364896      0   1.00000 1.00031 0.076053
## 2  0.154171      1   0.63510 0.66332 0.041423
## 3  0.122205      2   0.48093 0.50967 0.039603
## 4  0.046499      3   0.35873 0.39159 0.025264
## 5  0.034639      4   0.31223 0.33271 0.024717
## 6  0.028503      5   0.27759 0.31033 0.023712
## 7  0.023137      6   0.24909 0.28734 0.021780
## 8  0.017312      7   0.22595 0.25816 0.018958
## 9  0.016857      8   0.20864 0.23929 0.014902
## 10 0.013063      9   0.19178 0.22579 0.014091
## 11 0.011405     10   0.17872 0.21455 0.015026
## 12 0.010000     11   0.16731 0.20633 0.014531
```


Apparent
X Relative
R-square
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
Number of Splits


X Relative Error
0.2
0.4
0.6
0.8
1.0
1.2
0
2
4
6
8
10
Number of Splits

```
metrics_op = c(rmse_op,rsq_op[1],mae_op)
metrics_op
```

```
##           rel error
## 2.1568766 0.8326873 1.2141927
```

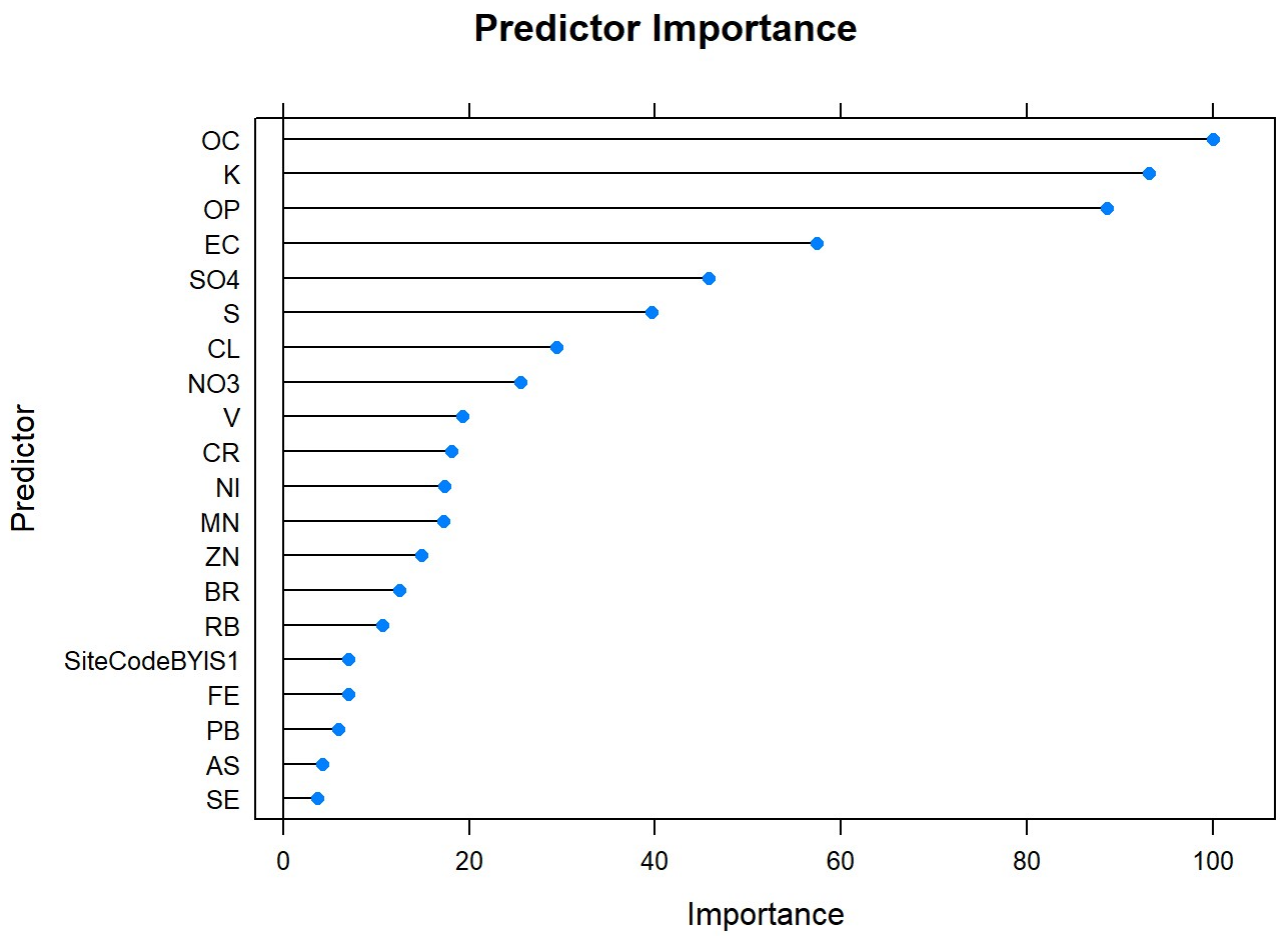
```
colnames(metrics_op)
```

```
## NULL
```

```
# Specify 10-fold cross validation
ctrl <- trainControl(method = "cv",  number = 10)

# CV bagged model
bagged_cv <- train(
  PM2.5 ~ .-PM2.5_UNC,
  data    = US_DATA_LRG,
  method = "treebag",
  trControl = ctrl,
  importance = TRUE
  )

# assess results
bagged_cv  #this is an object with many useful items
```

```
## Bagged CART
##
## 8647 samples
##   33 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 7781, 7783, 7781, 7783, 7783, 7783, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1.793048  0.8762557  0.9628197
```

```
# plot most important variables
plot(varImp(bagged_cv),20, main="Predictor Importance", ylab="Predictor")
```




```
varImp(bagged_cv)
```

```
## treebag variable importance
##
##   only 20 most important variables shown (out of 321)
##
##               Overall
## OC            100.000
## K              93.120
## OP             88.612
## EC             57.398
## SO4            45.777
## S              39.605
## CL             29.379
## NO3            25.574
## V              19.323
## CR             18.043
## NI             17.345
## MN             17.256
## ZN             14.872
## BR             12.469
## RB             10.654
## SiteCodeBYIS1   7.050
## FE              6.964
## PB              5.935
## AS              4.208
## SE              3.699
```

```
metric_bag= bagged_cv$results[1,][2:4]
bagged_cv
```

```
## Bagged CART
##
## 8647 samples
##   33 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 7781, 7783, 7781, 7783, 7783, 7783, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1.793048  0.8762557  0.9628197
```

```
metric_bag
```

```
##       RMSE  Rsquared       MAE
## 1 1.793048 0.8762557 0.9628197
```

```
metrics_fin = rbind(metric_bag,metrics_op)

rownames(metrics_fin)=c('Bagged_Tree_10cv','Optimal_GrdSrh_Tree_10cv' )
metrics_fin
```

```
##                               RMSE  Rsquared       MAE
## Bagged_Tree_10cv          1.793048 0.8762557 0.9628197
## Optimal_GrdSrh_Tree_10cv 2.156877 0.8326873 1.2141927
```