Code

Section 1: Summary statistics

```r
library(data.table)

pm = fread("C:/Users/nrt2117/Downloads/PRSA_data_2010.1.1-2014.12.31.csv")
dim(pm)
na_check = apply(is.na(pm),2,sum)
na_check

del = which(is.na(pm$pm2.5) == TRUE)
pm_delete_na = pm[-del,]
nrow(pm) - nrow(pm_delete_na)
apply(is.na(pm_delete_na),2,sum)
pm25 = pm_delete_na[,-1]

range(pm25$pm2.5)

summary(pm25$pm2.5)
summary(pm25$DEWP)
summary(pm25$TEMP)
summary(pm25$PRES)
summary(pm25$Iws)
table(pm25$cbwd)

#Separating pm2.5 data into levels according to WHO guide
pm25$index[pm25$pm2.5<=50]<-"Good"
pm25$index[pm25$pm2.5>=51 & pm25$pm2.5<=100]<-"Moderate"
pm25$index[pm25$pm2.5>=101 & pm25$pm2.5<=150]<-"Unhealthy for Sensitive Groups"
pm25$index[pm25$pm2.5>=151 & pm25$pm2.5<=100]<-"Unhealthy"
pm25$index[pm25$pm2.5>=201 & pm25$pm2.5<=300]<-"Very Unhealthy"
pm25$index[pm25$pm2.5>=301 & pm25$pm2.5<=500]<-"Hazardous"
pm25$index[pm25$pm2.5>500]<-"Beyond Index"

#See distribution according to levels
table(pm25$index)
counts <- table(pm25$index)
barplot(counts,main="PM2.5 Level Distribution",xlab="Levels")

#Some graphical representation
install.packages("ggubr")
library(ggubr)
gghistogram(pm,x="pm2.5", bins=50,add = "mean")

#Seasonal distribution
january <- pm25 %>%
  filter(month=="1") %>%
  select(pm2.5,index)
february <- pm25 %>%
  filter(month=="2") %>%
  select(pm2.5,index)
march <- pm25 %>%
  filter(month=="3") %>%
  select(pm2.5,index)
april <- pm25 %>%
  filter(month=="4") %>%
  select(pm2.5,index)
may <- pm25 %>%
  filter(month=="5") %>%
  select(pm2.5,index)
june <- pm25 %>%
  filter(month=="6") %>%
  select(pm2.5,index)
july <- pm25 %>%
  filter(month=="7") %>%
  select(pm2.5,index)
august <- pm25 %>%
  filter(month=="8") %>%
  select(pm2.5,index)
september <- pm25 %>%
  filter(month=="9") %>%
  select(pm2.5,index)
october <- pm25 %>%
  filter(month=="10") %>%
  select(pm2.5,index)
november <- pm25 %>%
  filter(month=="11") %>%
  select(pm2.5,index)
december <- pm25 %>%
  filter(month=="12") %>%
  select(pm2.5,index)

jancounts <- table(january$index)
barplot(jancounts,main="PM2.5 Level Distribution in January",xlab="Levels")
febcounts <- table(february$index)
barplot(febcounts,main="PM2.5 Level Distribution in February",xlab="Levels")
marcounts <- table(march$index)
barplot(marcounts,main="PM2.5 Level Distribution in March",xlab="Levels")
aprcounts <- table(april$index)
barplot(aprcounts,main="PM2.5 Level Distribution in April",xlab="Levels")
maycounts <- table(may$index)
barplot(maycounts,main="PM2.5 Level Distribution in May",xlab="Levels")
juncounts <- table(june$index)
barplot(juncounts,main="PM2.5 Level Distribution in June",xlab="Levels")
julcounts <- table(july$index)
barplot(julcounts,main="PM2.5 Level Distribution in July",xlab="Levels")
augcounts <- table(august$index)
barplot(augcounts,main="PM2.5 Level Distribution in August",xlab="Levels")
sepcounts <- table(september$index)
barplot(sepcounts,main="PM2.5 Level Distribution in September",xlab="Levels")
octcounts <- table(october$index)
barplot(octcounts,main="PM2.5 Level Distribution in October",xlab="Levels")
novcounts <- table(november$index)
```

```r
barplot(novcounts,main="PM2.5 Level Distribution in November",xlab="Levels")
deccounts <- table(december$index)
barplot(deccounts,main="PM2.5 Level Distribution in December",xlab="Levels")

#Look at 24 hour distribution
hour0 <- pm25 %>%
  filter(hour=="0") %>%
  select(pm2.5,index)
hour1 <- pm25 %>%
  filter(hour=="1") %>%
  select(pm2.5,index)
hour2 <- pm25 %>%
  filter(hour=="2") %>%
  select(pm2.5,index)
hour3 <- pm25 %>%
  filter(hour=="3") %>%
  select(pm2.5,index)
hour4 <- pm25 %>%
  filter(hour=="4") %>%
  select(pm2.5,index)
hour5 <- pm25 %>%
  filter(hour=="5") %>%
  select(pm2.5,index)
hour6 <- pm25 %>%
  filter(hour=="6") %>%
  select(pm2.5,index)
hour7 <- pm25 %>%
  filter(hour=="7") %>%
  select(pm2.5,index)
hour8 <- pm25 %>%
  filter(hour=="8") %>%
  select(pm2.5,index)
hour9 <- pm25 %>%
  filter(hour=="9") %>%
  select(pm2.5,index)
hour10 <- pm25 %>%
  filter(hour=="10") %>%
  select(pm2.5,index)
hour11 <- pm25 %>%
  filter(hour=="11") %>%
  select(pm2.5,index)
hour12 <- pm25 %>%
  filter(hour=="12") %>%
  select(pm2.5,index)
hour13 <- pm25 %>%
  filter(hour=="13") %>%
  select(pm2.5,index)
hour14 <- pm25 %>%
  filter(hour=="14") %>%
  select(pm2.5,index)
hour15 <- pm25 %>%
  filter(hour=="15") %>%
  select(pm2.5,index)
```

```r
hour16 <- pm25 %>%
  filter(hour=="16") %>%
  select(pm2.5,index)
hour17 <- pm25 %>%
  filter(hour=="17") %>%
  select(pm2.5,index)
hour18 <- pm25 %>%
  filter(hour=="18") %>%
  select(pm2.5,index)
hour19 <- pm25 %>%
  filter(hour=="19") %>%
  select(pm2.5,index)
hour20 <- pm25 %>%
  filter(hour=="20") %>%
  select(pm2.5,index)
hour21 <- pm25 %>%
  filter(hour=="21") %>%
  select(pm2.5,index)
hour22 <- pm25 %>%
  filter(hour=="22") %>%
  select(pm2.5,index)
hour23 <- pm25 %>%
  filter(hour=="23") %>%
  select(pm2.5,index)

hour0counts <- table(hour0$index)
barplot(hour0counts,main="PM2.5 Level Distribution in Hour0",xlab="Levels")
hour1counts <- table(hour1$index)
barplot(hour1counts,main="PM2.5 Level Distribution in Hour1",xlab="Levels")
hour2counts <- table(hour2$index)
barplot(hour2counts,main="PM2.5 Level Distribution in Hour2",xlab="Levels")
hour3counts <- table(hour3$index)
barplot(hour0counts,main="PM2.5 Level Distribution in Hour3",xlab="Levels")
hour4counts <- table(hour4$index)
barplot(hour4counts,main="PM2.5 Level Distribution in Hour4",xlab="Levels")
hour5counts <- table(hour5$index)
barplot(hour5counts,main="PM2.5 Level Distribution in Hour5",xlab="Levels")
hour6counts <- table(hour6$index)
barplot(hour6counts,main="PM2.5 Level Distribution in Hour6",xlab="Levels")
hour7counts <- table(hour7$index)
barplot(hour7counts,main="PM2.5 Level Distribution in Hour7",xlab="Levels")
hour8counts <- table(hour8$index)
barplot(hour8counts,main="PM2.5 Level Distribution in Hour8",xlab="Levels")
hour9counts <- table(hour9$index)
barplot(hour9counts,main="PM2.5 Level Distribution in Hour9",xlab="Levels")
hour10counts <- table(hour10$index)
barplot(hour10counts,main="PM2.5 Level Distribution in Hour10",xlab="Levels")
hour11counts <- table(hour11$index)
barplot(hour11counts,main="PM2.5 Level Distribution in Hour11",xlab="Levels")
hour12counts <- table(hour12$index)
barplot(hour12counts,main="PM2.5 Level Distribution in Hour12",xlab="Levels")
hour13counts <- table(hour13$index)
barplot(hour13counts,main="PM2.5 Level Distribution in Hour13",xlab="Levels")
```

```
hour14counts <- table(hour14$index)
barplot(hour14counts,main="PM2.5 Level Distribution in Hour14",xlab="Levels")
hour15counts <- table(hour15$index)
barplot(hour15counts,main="PM2.5 Level Distribution in Hour15",xlab="Levels")
hour16counts <- table(hour16$index)
barplot(hour16counts,main="PM2.5 Level Distribution in Hour16",xlab="Levels")
hour17counts <- table(hour17$index)
barplot(hour17counts,main="PM2.5 Level Distribution in Hour17",xlab="Levels")
hour18counts <- table(hour18$index)
barplot(hour18counts,main="PM2.5 Level Distribution in Hour18",xlab="Levels")
hour19counts <- table(hour19$index)
barplot(hour19counts,main="PM2.5 Level Distribution in Hour19",xlab="Levels")
hour20counts <- table(hour20$index)
barplot(hour20counts,main="PM2.5 Level Distribution in Hour20",xlab="Levels")
hour21counts <- table(hour21$index)
barplot(hour21counts,main="PM2.5 Level Distribution in Hour21",xlab="Levels")
hour22counts <- table(hour22$index)
barplot(hour22counts,main="PM2.5 Level Distribution in Hour22",xlab="Levels")
hour23counts <- table(hour23$index)
barplot(hour23counts,main="PM2.5 Level Distribution in Hour23",xlab="Levels")
```

Section 2: Time series analysis

```
library(data.table)
library(tseries)
library(itsmr)
library(forecast)
library(dplyr)
library(plyr)
path = "~/Desktop/GR5223 Multivariate Stat Inference/Beijing-PM-2.5-pullution-analysis/data"
pm = fread(paste0(path,"/PRSA_data_2010.1.1-2014.12.31.csv"))
na_check = apply(is.na(pm),2,sum);na_check

del = which(is.na(pm$pm2.5) == TRUE)
pm[del,"pm2.5"] = 0
pm25 = pm[,-1]
apply(is.na(pm25),2,sum);na_check

pm25_time = as.data.frame(pm25[,c(1,2,3,4,5)])
aggregate_pm = function(x){
  return(mean(x$pm2.5))
}
pm_daily  = ddply(pm25_time, .(year,month,day), aggregate_pm)
colnames(pm_daily) = c("year","month","day","pm2.5")
pm_daily = pm_daily$pm2.5


pm_monthly = ddply(pm25_time, .(year,month), aggregate_pm)
colnames(pm_monthly) = c("year","month","pm2.5")
pm_monthly = pm_monthly$pm2.5
```

Part1: Daily PM 2.5 values

```r
# time series plot
plot(1:length(pm_daily),ts(pm_daily),type="l",pch=22,lty=1,pty=2,col = "grey",
     ylab="PM2.5 concentration(ug/m^3)",xlab = "")
abline(h=mean(pm_daily),col = "red", lwd = 3, lty = 4)

# logarithmic data
pm_daily = log(pm_daily[which(pm_daily != 0)])
plot(1:length(pm_daily),ts(pm_daily),type="l",pch=22,lty=1,pty=2,col="grey",
     ylab="PM2.5 concentration(ug/m^3)",xlab = "Figure 1-2")
abline(h=mean(pm_daily),col = "red", lwd = 3, lty = 4)

# ACF and PACF plots
op = par(mfrow=c(1,2))
acf(pm_daily,lag.max = 40,
         xlab = "time lag", ylab = 'ACF',main='ACF')
acf(pm_daily,lag.max = 40, type = "partial",
         xlab = "time lag", ylab = 'PACF', main='PACF')
par(op)

# fit model
fit_daily = auto.arima(ts(pm_daily))
fit_daily$coef

# residuals
daily_res = fit_daily$residuals

# Test residuals
Box.test(daily_res,lag=20,type="Box-Pierce") #p-value = 0.9885

qqnorm(daily_res); qqline(daily_res)
shapiro.test(daily_res)   #p-value < 2.2e-16

matrix(data = c(round(mean(daily_res),5),round(var(daily_res),5)),
       nrow = 1, byrow = F, dimnames = list(c("Value"),c("Mean","Variance")))
plot(daily_res, type = "l",
     main = "Residuals of fitted MA(2) model",
     xlab='', ylab='Residuals')

# Test the stationarity of the PM2.5 data
Box.test(pm_daily,lag=20,type="Box-Pierce") #p-value < 2.2e-16

shapiro.test(pm_daily)   #p-value = 1.261e-12

adf_daily = adf.test(pm_daily)  #p-value = 0.01
adf_daily
```

Part 2: Monthly PM 2.5 values

```r
# time series plot of original monthly data and logarithmic data
op = par(mfrow=c(1,2))
```

```
plot(1:length(pm_monthly),ts(pm_monthly),type="l",pch=22,lty=1,pty=2,
     ylab="PM2.5 concentration(ug/m^3)",xlab = "Monthly data")
abline(h=mean(pm_monthly),col = "red", lwd = 3, lty = 4)

pm_monthly = log(pm_monthly[which(pm_monthly != 0)])
plot(1:length(pm_monthly),ts(pm_monthly),type="l",pch=22,lty=1,pty=2,
     ylab="PM2.5 concentration(ug/m^3)",xlab = "log Monthly data")
abline(h=mean(pm_monthly),col = "red", lwd = 3, lty = 4)
par(op)

op = par(mfrow=c(1,2))
# the stationary signal and ACF
acf(pm_monthly,lag.max = 40,
         xlab = "lag #", ylab = 'ACF',main=' ')
# the trend signal and ACF
acf(pm_monthly,lag.max = 40, type = "partial",
         xlab = "lag #", ylab = 'PACF', main=' ')
par(op)

# Fit 2 models
fit_monthly_ma = Arima(pm_monthly,order=c(0,0,6))
fit_monthly_ar = Arima(pm_monthly,order=c(6,0,0))

matrix(data = round(c(fit_monthly_ma$aic,fit_monthly_ar$aic),3),
       nrow = 1, ncol = 2,
       dimnames = list("AIC",c("MA(6)","AR(6)")))
fit_monthly_ar$coef

# Test residuals
monthly_res = fit_monthly_ma$residuals
Box.test(monthly_res,lag=20,type="Box-Pierce") # p-value = 0.9851

shapiro.test(monthly_res)   #p-value = 0.4591

matrix(data = c(round(mean(monthly_res),5),round(var(monthly_res),5)),
       byrow = F, nrow = 1, dimnames = list(c("value"),c("Mean","Variance")))

Box.test(pm_monthly,lag=20,type="Box-Pierce") #p-value = 0.1197

shapiro.test(pm_monthly)   #p-value = 0.8307

adf_monthly = adf.test(pm_monthly)  #p-value = 0.04378
adf_monthly

# Compare models
fit_daily_error = summary(fit_daily)
fit_monthly_error = summary(fit_monthly_ma)
fit_daily_rmse = fit_daily_error[2]
fit_monthly_rmse = fit_monthly_error[2]

matrix(data = round(c(fit_daily$aic,fit_monthly_ma$aic,
                fit_daily_rmse,fit_monthly_rmse),3),
       nrow = 2, byrow = F,
       dimnames = list(c("Daily PM 2.5","Monthly PM 2.5"),c("AIC","RMSE")))
```

Section 3: Linear regression analysis

```
names(pm25)
X = pm25[,c("PRES","DEWP","TEMP","Is","Ir","Iws")]
W = pm25$cbwd
unique(W)
W[which(W == "NW")] = 1
W[which(W == "NE")] = 2
W[which(W == "SE")] = 3
W[which(W == "cv")] = 4

pm_data = data.frame(pm25$pm2.5,X,Wind_dir = W)
colnames(pm_data) = c("PM2.5","PRES","DEWP","TEMP","Snow","Rain","Wind_speed","Wind_dir")
pm_data$Wind_dir = as.factor(pm_data$Wind_dir)
tail(pm_data)
names(pm_data)
```

Time series error

```
pm_ts = ts(pm_data)
fit = auto.arima(pm_ts[,1], xreg = as.matrix(pm_ts[,2:7]))
print(fit)
plot(ts(pm_data$PM2.5))

cbind("Regression Errors" = residuals(fit, type="regression"),
      "ARIMA errors" = residuals(fit, type="innovation")) %>%
  autoplot(facets=TRUE)
checkresiduals(fit)
# Not a white noise, which is matched with the claim in the reference paper
#set.seed(2019)
#residules=arima.sim(model=list(ar=0.5,ma=c(0.1804,-0.0075)),n=nrow(pm_data))

forecast_residule = forecast(fit, xreg=as.matrix(pm_ts[,2:7]))
```

Fit regression model part

```
#transformation of predictor variable y

model_1 = lm(PM2.5~.,data = pm_data)
sum_1 = summary(model_1)
sum_1$adj.r.squared
AIC(model_1)
qqnorm(rstudent(model_1));qqline(rstudent(model_1))


pm_data_log = pm_data[which(pm_data$PM2.5 != 0),]
log_data = pm_data_log
log_data$PM2.5 = log(log_data$PM2.5)
```

```r
#sum(pm_data$PM2.5 == 0)
model_2 = lm(PM2.5~.,data = log_data)
sum_2 = summary(model_2)
sum_2$coefficients
sum_2$adj.r.squared
AIC(model_2)
BIC(model_2)
qqnorm(rstudent(model_2));qqline(rstudent(model_2))
```

```r
data_matrix = model.matrix(PM2.5~., log_data)[,-1]
lambda = 10^seq(-5,10, length = 200)

ridge_fit = cv.glmnet(data_matrix, log_data$PM2.5, alpha = 0, lambda = lambda)
model_ridge = glmnet(data_matrix, log_data$PM2.5,lambda = lambda)

plot(ridge_fit)
opt_lambda_r = ridge_fit$lambda.min
opt_lambda_r

ridge_coe = predict(model_ridge, type = "coefficients", s = opt_lambda_r )
ridge_coe
# Compute R^2
y_pred = predict(model_ridge, s = opt_lambda_r, newx = data_matrix)
# Sum of Squares Total and Error
sst = sum((log_data$PM2.5 - mean(log_data$PM2.5))^2)
sse = sum((y_pred - log_data$PM2.5)^2)
r_2 = 1 - sse / sst
r_2
sum_2$adj.r.squared
# Slightly improve


lasso_fit = cv.glmnet(data_matrix, log_data$PM2.5,alpha = 1, lambda = lambda)
model_lasso = glmnet(data_matrix, log_data$PM2.5, alpha = 1, lambda = lambda)
opt_lambda_l = lasso_fit$lambda.min
opt_lambda_l

plot(model_lasso, label = T)
lasso_coef = predict(model_lasso, type = "coefficients", s = opt_lambda_l )
lasso_coef
sum_2$coefficients
```

```r
library(leaps)
library(bestglm)
log_data1 = log_data[,c(2:8,1)]
#levels(log_data$Wind_dir)
bestglm(log_data1,IC="AIC")
bestglm(log_data1,IC="BIC")
min.model=lm(PM2.5 ~ 1,data=log_data1)
full.model=formula(model_2)
step(min.model,scope=full.model,direction=c("forward"))
step(min.model,scope=full.model,direction=c("backward"))
```

```
best_subset = regsubsets(PM2.5~., data = log_data1, method = "exhaustive")
forward = regsubsets(PM2.5~., data = log_data1, method = "forward")
backward = regsubsets(PM2.5~., data = log_data1, method = "backward")
best = summary(best_subset)
forw = summary(forward)
back = summary(backward)

plot(x = 1:8, y = best$rss, col = 2, pch = 2, type = "o",
     main = "Subset selection", xlab = "Number of variables", ylab = "RSS")
points(x = 1:8, y = forw$rss, col = 3, pch = 3, type = "o")
points(x = 1:8, y = back$rss, col = 4, pch = 4, type = "o")
legend("topright", legend = c("Best subset", "Forward", "Backward"),
       col = c(2,3,4), pch = c(2,3,4))


op = par(mfrow = c(1,2))
plot(x = 1:8, y = best$cp, type = "o", col = 2,
     xlab = "Number of variables", ylab = "Cp" )
plot(x = 1:8, y = best$bic, type = "o", col = 3,
     xlab = "Number of variables", ylab = "BIC" )
par(op)
which.min(best$cp)
which.min(best$bic)
best$outmat


op = par(mfrow = c(1,2))
plot(x = 1:8, y = forw$cp, type = "o", col = 2,
     xlab = "Number of variables", ylab = "Cp" )
plot(x = 1:8, y = forw$bic, type = "o", col = 3,
     xlab = "Number of variables", ylab = "BIC" )
par(op)
which.min(forw$cp)
which.min(forw$bic)
forw$outmat


op = par(mfrow = c(1,2))
plot(x = 1:8, y = back$cp, type = "o", col = 2,
     xlab = "Number of variables", ylab = "Cp" )
plot(x = 1:8, y = back$bic, type = "o", col = 3,
     xlab = "Number of variables", ylab = "BIC" )
par(op)
which.min(back$cp)
which.min(back$bic)
back$outmat
```

```
#relationships among quantitative variables
#library(GGally)
cor(log_data[,-c(1,8)])
library(car)
vif(model_2)
# multicolinearity exists
# consider to use shrinkage method
```

```r
names(log_data) = c("Y","X1","X2","X3","X4","X5","X6","X7")
names(pm_data)
model_3 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X1*X2), data = log_data)  # pressure(X1) and dew(X2)
sum_3 = summary(model_3)
sum_2$coefficients
sum_3$coefficients      # p-values of X1 and X2 are lower, and the interaction is significant
sum_2$adj.r.squared     #0.413282
sum_3$adj.r.squared     #0.434887, doesn't have much improvement
AIC(model_2) #99005
AIC(model_3) #97439 the value of AIC decreases but not much


model_4 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X2*X3), data = log_data)  # Dew(X2) and temp(X3)
sum_4 = summary(model_4)
sum_4$coefficients      # p-values of X1 and X2 are lower, and the interaction is significant
sum_4$adj.r.squared     #0.4250937, doesn't have much improvement, less better than model_3
AIC(model_4) #98156 less better than model_3


model_5 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X2*X4), data = log_data)  # Dew(X2) and Snow(X4)
sum_5 = summary(model_5)
sum_5$adj.r.squared     #0.41349, has no improvement,
AIC(model_5) #98991 less better than model_3

model_6 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X2*X5)+I(X4*X5), data = log_data)  # Dew(X2) and rain(X4), snow a
sum_6 = summary(model_6)
sum_6$adj.r.squared     #0.413282 has no improvement
AIC(model_6) #99000 no improvement

model_7 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X1*X6), data = log_data)  # pressure(X1) and wind speed(X6)
sum_7 = summary(model_7)
sum_7$adj.r.squared     #0.4150446 has no improvement
AIC(model_7) #98880 not much improvement

model_8 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X1*X3), data = log_data) #pressure(X1) and temp(X3)
sum_8 = summary(model_8)
sum_8$adj.r.squared     #0.422 not much improvement
AIC(model_8) # 98370 not much improvement

model_9 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X1*X2)+I(X1*X3), data = log_data) #pressure(X1) and temp(X3)
sum_9 = summary(model_9)
sum_9$adj.r.squared     #0.436 not much improvement
AIC(model_9) #97325 not much improvement

# No interaction need to be added to the model
```

```r
model_3 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X1*X1)+I(X2*X2)+I(X3*X3)+I(X4*X4)+I(X5*X5)+I(X6*X6), data = log_da
sum_3 = summary(model_3)
#sum_2$coefficients
sum_3$coefficients      # p-values of X1 and X2 are lower, and the interaction is significant
sum_2$adj.r.squared     #0.413282
sum_3$adj.r.squared     #0.4410534, doesn't have much improvement

model_4 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X1*X1), data = log_data)  # pressure(X1) and dew(X2)
```

```
sum_4 = summary(model_4)
sum_4$adj.r.squared     #0.4278366, doesn't have much improvement

model_5 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X2*X2), data = log_data)  # pressure(X1) and dew(X2)
sum_5 = summary(model_5)
sum_5$adj.r.squared     # 0.4229174, doesn't have much improvement


model_6 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X3*X3), data = log_data)  # pressure(X1) and dew(X2)
sum_6 = summary(model_6)
sum_6$adj.r.squared     # 0.4146235, doesn't have much improvement

model_7 = lm(Y~X1+X2+X3+X4+X5+X6+X7+I(X4*X4), data = log_data)  # pressure(X1) and dew(X2)
sum_7 = summary(model_7)
sum_7$adj.r.squared     # 0.4133083, doesn't have much improvement
```

```
#y and quantitative variables:

# y and X1(pres)
# With line
boxplot(Y~X1, data = log_data)
abline(lm(Y ~ X1, data = log_data),col="purple")

# With smoother
boxplot(Y~X1, data = log_data)
lines(supsmu(log_data$X1,log_data$Y),col="purple")

poly.model.1 <- lm(Y~X1+X2+X3+X4+X5+X6+X7+poly(X1,2),data = log_data)
sum_poly1 <- summary(poly.model.1)
sum_poly1$adj.r.squared #0.428 doesn't have much improvement
AIC(poly.model.1) # 97957 doesn't have much improvement

# y and X2(dew)
# With line
boxplot(Y~X2, data = log_data)
abline(lm(Y~X2, data = log_data),col="purple")

# With smoother
boxplot(Y~X2, data = log_data)
lines(supsmu(log_data$X2,log_data$Y),col="purple")

poly.model.2 <- lm(Y~X1+X2+X3+X4+X5+X6+X7+sqrt(abs(X2)),data = log_data)
sum_poly2 <- summary(poly.model.2)
sum_poly2$adj.r.squared #0.424 doesn't have much improvement
AIC(poly.model.2)# 98240 doesn't have uch improvement

# y and X3(temp)
# With line
boxplot(Y~X3, data = log_data)
abline(lm(Y~X3, data = log_data),col="purple")

# With smoother
boxplot(Y~X3, data = log_data)
lines(supsmu(log_data$X3,log_data$Y),col="purple")
```

```
poly.model.3 <- lm(Y~X1+X2+X3+X4+X5+X6+X7+poly(X3,2),data = log_data)
sum_poly3 <- summary(poly.model.3)
sum_poly3$adj.r.squared #0.415 doesn't have much improvement
AIC(poly.model.3) # 98910 doesn't have much improvement

# y and X4(snow)
# With line
boxplot(Y~X4, data = log_data)
abline(lm(Y~X4, data = log_data),col="purple")

# With smoother
boxplot(Y~X4, data = log_data)
lines(supsmu(log_data$X4,log_data$Y),col="purple")

# y and X5(rain)
# With line
boxplot(Y~X5, data = log_data)
abline(lm(Y~X5, data = log_data),col="purple")

# With smoother
boxplot(Y~X5, data = log_data)
lines(supsmu(log_data$X5,log_data$Y),col="purple")

# y and X6(wind speed)
# With line
boxplot(Y~X6, data = log_data)
abline(lm(Y~X6, data = log_data),col="purple")

# With smoother
boxplot(Y~X6, data = log_data)
lines(supsmu(log_data$X6,log_data$Y),col="purple")
```

```
#relationship between quantitative variabels and qualitative variables
# First define logical variables
nw <- pm25$cbwd=="NW"
cv <- pm25$cbwd=="cv"
se <- pm25$cbwd=="SE"
ne <- pm25$cbwd=="NE"

#X1(pres) and X7(cbwd)
# Scatter plot with smoothers for each cbwd level
plot(log_data$X1,log_data$Y,col="lightgrey",xlab="pres",ylab="log(pm2.5)")
abline(lm((log_data$Y)[nw]~log_data$X1[nw]),col=2)
abline(lm((log_data$Y)[cv]~log_data$X1[cv]),col=3)
abline(lm((log_data$Y)[se]~log_data$X1[se]),col=4)
abline(lm((log_data$Y)[ne]~log_data$X1[ne]),col=5)
legend("topleft",legend=c("nw","cv","se","ne"),fill=2:5)

inter.model.1 <- lm(Y~X1+X2+X3+X4+X5+X6+X7+X1*X7,data=log_data)
summary(inter.model.1)
sum_inter1<-summary(inter.model.1)
sum_inter1$adj.r.squared #0.416 doesn't have improvement
AIC(inter.model.1)# 98790 dpesn't have improvment
```

```r
#X2(dewp) and X7(cbwd)
plot(log_data$X2,log_data$Y,col="lightgrey",xlab="dewp",ylab="log(pm2.5)")
abline(lm((log_data$Y)[nw]~log_data$X2[nw]),col=2)
abline(lm((log_data$Y)[cv]~log_data$X2[cv]),col=3)
abline(lm((log_data$Y)[se]~log_data$X2[se]),col=4)
abline(lm((log_data$Y)[ne]~log_data$X2[ne]),col=5)
legend("topleft",legend=c("nw","cv","se","ne"),fill=2:5)

inter.model.2 <- lm(Y~X1+X2+X3+X4+X5+X6+X7+X2*X7,data=log_data)
summary(inter.model.2)
sum_inter2<-summary(inter.model.2)
sum_inter2$adj.r.squared #0.415 doesn't have improvement
AIC(inter.model.2)# 98861 dpesn't have improvment

#X3(temp) and X7(cbwd)
plot(log_data$X3,log_data$Y,col="lightgrey",xlab="temp",ylab="log(pm2.5)")
abline(lm((log_data$Y)[nw]~log_data$X3[nw]),col=2)
abline(lm((log_data$Y)[cv]~log_data$X3[cv]),col=3)
abline(lm((log_data$Y)[se]~log_data$X3[se]),col=4)
abline(lm((log_data$Y)[ne]~log_data$X3[ne]),col=5)
legend("topleft",legend=c("nw","cv","se","ne"),fill=2:5)

inter.model.3 <- lm(Y~X1+X2+X3+X4+X5+X6+X7+X3*X7,data=log_data)
summary(inter.model.3)
sum_inter3<-summary(inter.model.3)
sum_inter3$adj.r.squared #0.414 doesn't have improvement
AIC(inter.model.3)# 98929 dpesn't have improvment

#X4(snow) and X7(cbwd)
plot(log_data$X3,log_data$Y,col="lightgrey",xlab="snow",ylab="log(pm2.5)")
abline(lm((log_data$Y)[nw]~log_data$X4[nw]),col=2)
abline(lm((log_data$Y)[cv]~log_data$X4[cv]),col=3)
abline(lm((log_data$Y)[se]~log_data$X4[se]),col=4)
abline(lm((log_data$Y)[ne]~log_data$X4[ne]),col=5)
legend("topleft",legend=c("nw","cv","se","ne"),fill=2:5)

inter.model.4 <- lm(Y~X1+X2+X3+X4+X5+X6+X7+X4*X7,data=log_data)
summary(inter.model.4)
sum_inter4<-summary(inter.model.4)
sum_inter4$adj.r.squared #0.414 doesn't have improvement
AIC(inter.model.4)# 98965 dpesn't have improvment

#X5(rain) and X7(cbwd)
plot(log_data$X5,log_data$Y,col="lightgrey",xlab="rain",ylab="log(pm2.5)")
abline(lm((log_data$Y)[nw]~log_data$X5[nw]),col=2)
abline(lm((log_data$Y)[cv]~log_data$X5[cv]),col=3)
abline(lm((log_data$Y)[se]~log_data$X5[se]),col=4)
abline(lm((log_data$Y)[ne]~log_data$X5[ne]),col=5)
legend("topleft",legend=c("nw","cv","se","ne"),fill=2:5)

inter.model.5 <- lm(Y~X1+X2+X3+X4+X5+X6+X7+X5*X7,data=log_data)
summary(inter.model.5)
sum_inter5<-summary(inter.model.5)
```

```r
sum_inter5$adj.r.squared #0.415 doesn't have improvement
AIC(inter.model.5)# 99005 dpesn't have improvment

#X6(wind speed) and X7(cbwd)
plot(log_data$X6,log_data$Y,col="lightgrey",xlab="wind speed",ylab="log(pm2.5)")
abline(lm((log_data$Y)[nw]~log_data$X6[nw]),col=2)
abline(lm((log_data$Y)[cv]~log_data$X6[cv]),col=3)
abline(lm((log_data$Y)[se]~log_data$X6[se]),col=4)
abline(lm((log_data$Y)[ne]~log_data$X6[ne]),col=5)
legend("topleft",legend=c("nw","cv","se","ne"),fill=2:5)

inter.model.6 <- lm(Y~X1+X2+X3+X4+X5+X6+X7+X6*X7,data=log_data)
summary(inter.model.6)
sum_inter6<-summary(inter.model.6)
sum_inter6$adj.r.squared #0.419 doesn't have improvement
AIC(inter.model.6)# 98596 dpesn't have improvment
```

model validation

```r
# Model validation

# Choose training and validation set

set.seed(0)
round(.2*nrow(log_data))
index <- sample(1:nrow(log_data),8351,replace = F)
train.data <- log_data[-index,]
data <- train.data
test.data <- log_data[index,]

# Compute MSPR

#Below we compute the MSPR using our final model trained from the training set on the test set. First f

bad.final.model <- lm(Y~X1+X2+X3+X4+X5+X6+X7,data= train.data)

# Compute MSE
MSE <- sum((residuals(bad.final.model))^2)/(nrow(train.data)-7)

# For comparison, we can compute MSE of the earlier final model
MSE.earler <- sum((residuals(model_2))^2)/(nrow(log_data)-7)

names(train.data[,-1])

Y.test <- test.data[,1]
X.test <- test.data[,-1]
n.test <- nrow(X.test)
n.test
Y.hat.test <- predict(bad.final.model,newdata = X.test)
length(Y.hat.test)==n.test


# MSPR
MSPR <- mean((Y.test-Y.hat.test)^2)
```

```
# Compare
round(c(MSPR=MSPR,MSE=MSE,MSEearler=MSE.earler),4)
```

plot of the final model

```
model.final<-model_2

boxplot(log_data$Y~log_data$X7,main="PM2.5 by wind direction",ylab="PM2.5")

qqnorm(rstudent(model.final),main="QQ-Plot")
abline(a=0,b=1,lty=3,col = "red")

plot(1:length(log_data$Y),rstudent(model.final),main="Line Plot", ylab="Deleted Residuals")
abline(h=0,lty=3)
lines(1:length(log_data$Y),rstudent(model.final), col = 2)

plot(predict(model.final),rstudent(model.final),main="Residual Plot",xlab="Y-hat",ylab="Deleted Residual
abline(h=0,lty=2)
lines(supsmu(predict(model.final),rstudent(model.final)),col=2)
```