Regression_Simon_Kim

Simon Kim, sxk190106

2022-09-25

What is linear regression

Regression: By dividing values into predictor and target, we can make a relationship between two values.

Linear regression: if the relationship from regression shapes linear, we call it as a linear regression. In other words, linear regression is finding a linear graph that best explains the given data.

Strengths 1. Learning speed is fast and prediction is fast 2. Works well on very large and sparse datasets 3. It is relatively easy to understand how predictions are made through formulas. 4. The advantage is that there are no parameters, but there is no way to control the complexity of the model.

Weakness 1. Limited to linear relationships, sometimes incorrect assuming that there is a linear relationship between them 2. Only the mean of the dependent variable is viewed, difficult when it is necessary to look at the extremes of the dependent variable, and linear regression is not a complete description of the relationship between variables, just as the mean is not a complete description of a single variable. Quaternary regression is used to troubleshoot problems 3. Sensitive to outliers 4. Linear regression assumes that the data are independent, but not always reasonable. Solvable using multi-level models 5. The values of the coefficients are sometimes unclear why. This is especially true when the characteristics of datasets are deeply annualized

Regression

The data for this activity is from here

The data is about power consumption of three different distribution networks of Tetouan city which is located in north Morocco..

Call data set

This code calls data from my computer and store it as df

In the given file, there are NA values. Therefore, I will change NA into 0

```
library(readr)
df <- read_csv("PRSA_data_2010.1.1-2014.12.31.csv")
```

```
## Rows: 43824 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): cbwd
## dbl (12): No, year, month, day, hour, pm2.5, DEWP, TEMP, PRES, Iws, Is, Ir
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df[is.na(df)]<-0
```

dividing

This code divides df into training and test the data set into 80/20 Train/Test by randomly sampling the rows.

```
set.seed(1234)
i <- sample(1:nrow(df), nrow(df)*0.80,
replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

data exploration

In this section, I will use 5 functions for each tain and test for data exploration

1. dim(): check number of observation and variables
2. head(): see the first 6 of the data set
3. str(): check data structure and type
4. summary(): check frequency of base statistics
5. apply(): check number of NA values of each columns by na using apply+2+sum

```
dim(train)
```

```
## [1] 35059    13
```

```
dim(test)
```

```
## [1] 8765   13
```

```
head(train)
```

```
## # A tibble: 6 x 13
##      No  year month   day  hour pm2.5  DEWP  TEMP  PRES cbwd    Iws    Is    Ir
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <dbl> <dbl>
## 1 40784  2014     8    27     7    21    15    19  1014 NW     4.92     0     0
## 2 40854  2014     8    30     5   154    20    21  1016 NE     1.79     0     0
## 3 41964  2014    10    15    11    11    -2    21  1014 NW    18.8      0     0
## 4 15241  2011     9    28     0   177    14    16  1015 SE     1.79     0     0
## 5 33702  2013    11     5     5   152     0     7  1022 SE     1.79     0     0
## 6 35716  2014     1    28     3    13   -21    -4  1029 NW    10.3      0     0
```

```
head(test)
```

```
## # A tibble: 6 x 13
##      No  year month   day  hour pm2.5  DEWP  TEMP  PRES cbwd    Iws    Is    Ir
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <dbl> <dbl>
## 1     4  2010     1     1     3     0   -21   -14  1019 NW     9.84     0     0
## 2     5  2010     1     1     4     0   -20   -12  1018 NW    13.0      0     0
## 3    22  2010     1     1    21     0   -17    -5  1018 NW     1.79     0     0
## 4    30  2010     1     2     5   109    -7    -6  1022 SE     7.14     3     0
## 5    36  2010     1     2    11   152    -8    -5  1026 SE    20.6      0     0
## 6    40  2010     1     2    15   154    -9    -5  1025 SE    35.8      0     0
```

```
str(train)
```

```
## tibble [35,059 x 13] (S3: tbl_df/tbl/data.frame)
##  $ No   : num [1:35059] 40784 40854 41964 15241 33702 ...
##  $ year : num [1:35059] 2014 2014 2014 2011 2013 ...
##  $ month: num [1:35059] 8 8 10 9 11 1 12 9 4 4 ...
##  $ day  : num [1:35059] 27 30 15 28 5 28 30 27 6 20 ...
##  $ hour : num [1:35059] 7 5 11 0 5 3 14 3 13 5 ...
```

```
##  $ pm2.5: num [1:35059] 21 154 11 177 152 13 150 226 35 87 ...
##  $ DEWP : num [1:35059] 15 20 -2 14 0 -21 -8 14 -8 7 ...
##  $ TEMP : num [1:35059] 19 21 21 16 7 -4 -2 14 21 8 ...
##  $ PRES : num [1:35059] 1014 1016 1014 1015 1022 ...
##  $ cbwd : chr [1:35059] "NW" "NE" "NW" "SE" ...
##  $ Iws  : num [1:35059] 4.92 1.79 18.77 1.79 1.79 ...
##  $ Is   : num [1:35059] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Ir   : num [1:35059] 0 0 0 0 0 0 0 0 0 0 ...
```

```
str(test)
```

```
## tibble [8,765 x 13] (S3: tbl_df/tbl/data.frame)
##  $ No   : num [1:8765] 4 5 22 30 36 40 41 44 45 48 ...
##  $ year : num [1:8765] 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ month: num [1:8765] 1 1 1 1 1 1 1 1 1 1 ...
##  $ day  : num [1:8765] 1 1 1 2 2 2 2 2 2 2 ...
##  $ hour : num [1:8765] 3 4 21 5 11 15 16 19 20 23 ...
##  $ pm2.5: num [1:8765] 0 0 0 109 152 154 159 149 154 126 ...
##  $ DEWP : num [1:8765] -21 -20 -17 -7 -8 -9 -9 -8 -7 -8 ...
##  $ TEMP : num [1:8765] -14 -12 -5 -6 -5 -5 -5 -5 -5 -6 ...
##  $ PRES : num [1:8765] 1019 1018 1018 1022 1026 ...
##  $ cbwd : chr [1:8765] "NW" "NW" "NW" "SE" ...
##  $ Iws  : num [1:8765] 9.84 12.97 1.79 7.14 20.56 ...
##  $ Is   : num [1:8765] 0 0 0 3 0 0 0 0 0 3 ...
##  $ Ir   : num [1:8765] 0 0 0 0 0 0 0 0 0 0 ...
```

```
summary(train)
```

```
##        No              year          month              day        
##  Min.   :    1   Min.   :2010   Min.   : 1.000   Min.   : 1.00  
##  1st Qu.:11044   1st Qu.:2011   1st Qu.: 4.000   1st Qu.: 8.00  
##  Median :21975   Median :2012   Median : 7.000   Median :16.00  
##  Mean   :21956   Mean   :2012   Mean   : 6.513   Mean   :15.76  
##  3rd Qu.:32930   3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00  
##  Max.   :43824   Max.   :2014   Max.   :12.000   Max.   :31.00  
##       hour           pm2.5          DEWP              TEMP        
##  Min.   : 0.00   Min.   :  0   Min.   :-40.000   Min.   :-19.00  
##  1st Qu.: 5.00   1st Qu.: 24   1st Qu.:-10.000   1st Qu.:  2.00  
##  Median :12.00   Median : 68   Median :  2.000   Median : 14.00  
##  Mean   :11.49   Mean   : 94   Mean   :  1.831   Mean   : 12.49  
##  3rd Qu.:17.00   3rd Qu.:133   3rd Qu.: 15.000   3rd Qu.: 23.00  
##  Max.   :23.00   Max.   :994   Max.   : 28.000   Max.   : 42.00  
##       PRES          cbwd                Iws                Is          
##  Min.   : 991   Length:35059       Min.   :  0.45   Min.   : 0.00000  
##  1st Qu.:1008   Class :character   1st Qu.:  1.79   1st Qu.: 0.00000  
##  Median :1016   Mode  :character   Median :  5.37   Median : 0.00000  
##  Mean   :1016                      Mean   : 23.60   Mean   : 0.04932  
##  3rd Qu.:1025                      3rd Qu.: 21.90   3rd Qu.: 0.00000  
##  Max.   :1046                      Max.   :585.60   Max.   :27.00000  
##        Ir         
##  Min.   : 0.0000  
##  1st Qu.: 0.0000  
##  Median : 0.0000  
##  Mean   : 0.1933  
##  3rd Qu.: 0.0000
```

```
##  Max.   :36.0000
```

```
summary(test)
```

```
##        No             year          month             day
##  Min.   :    4   Min.   :2010   Min.   : 1.000   Min.   : 1.00
##  1st Qu.:10549   1st Qu.:2011   1st Qu.: 4.000   1st Qu.: 8.00
##  Median :21692   Median :2012   Median : 7.000   Median :16.00
##  Mean   :21737   Mean   :2012   Mean   : 6.567   Mean   :15.61
##  3rd Qu.:32598   3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00
##  Max.   :43811   Max.   :2014   Max.   :12.000   Max.   :31.00
##       hour           pm2.5             DEWP              TEMP
##  Min.   : 0.00   Min.   :  0.00   Min.   :-37.000   Min.   :-18.00
##  1st Qu.: 6.00   1st Qu.: 24.00   1st Qu.:-10.000   1st Qu.:  1.00
##  Median :11.00   Median : 67.00   Median :  2.000   Median : 13.00
##  Mean   :11.54   Mean   : 93.83   Mean   :  1.764   Mean   : 12.29
##  3rd Qu.:18.00   3rd Qu.:132.00   3rd Qu.: 15.000   3rd Qu.: 23.00
##  Max.   :23.00   Max.   :972.00   Max.   : 27.000   Max.   : 41.00
##       PRES          cbwd                Iws               Is
##  Min.   : 992   Length:8765        Min.   :  0.45   Min.   : 0.0000
##  1st Qu.:1008   Class :character   1st Qu.:  1.79   1st Qu.: 0.0000
##  Median :1016   Mode  :character   Median :  5.37   Median : 0.0000
##  Mean   :1016                      Mean   : 25.04   Mean   : 0.0664
##  3rd Qu.:1025                      3rd Qu.: 22.79   3rd Qu.: 0.0000
##  Max.   :1046                      Max.   :573.54   Max.   :25.0000
##        Ir
##  Min.   : 0.0000
##  1st Qu.: 0.0000
##  Median : 0.0000
##  Mean   : 0.2014
##  3rd Qu.: 0.0000
##  Max.   :30.0000
```

```
apply(is.na(train), MARGIN = 2, FUN = 'sum')
```

```
##    No  year month   day  hour pm2.5  DEWP  TEMP  PRES  cbwd   Iws    Is    Ir
##     0     0     0     0     0     0     0     0     0     0     0     0     0
```

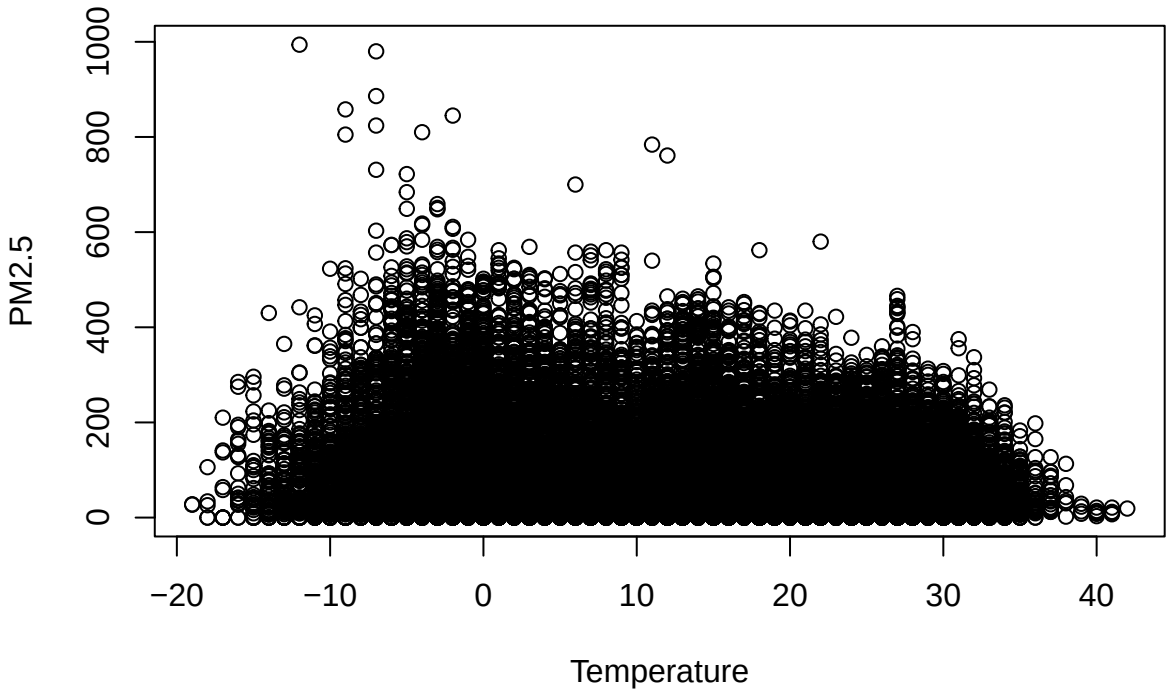
```
apply(is.na(test), MARGIN = 2, FUN = 'sum')
```

```
##    No  year month   day  hour pm2.5  DEWP  TEMP  PRES  cbwd   Iws    Is    Ir
##     0     0     0     0     0     0     0     0     0     0     0     0     0
```

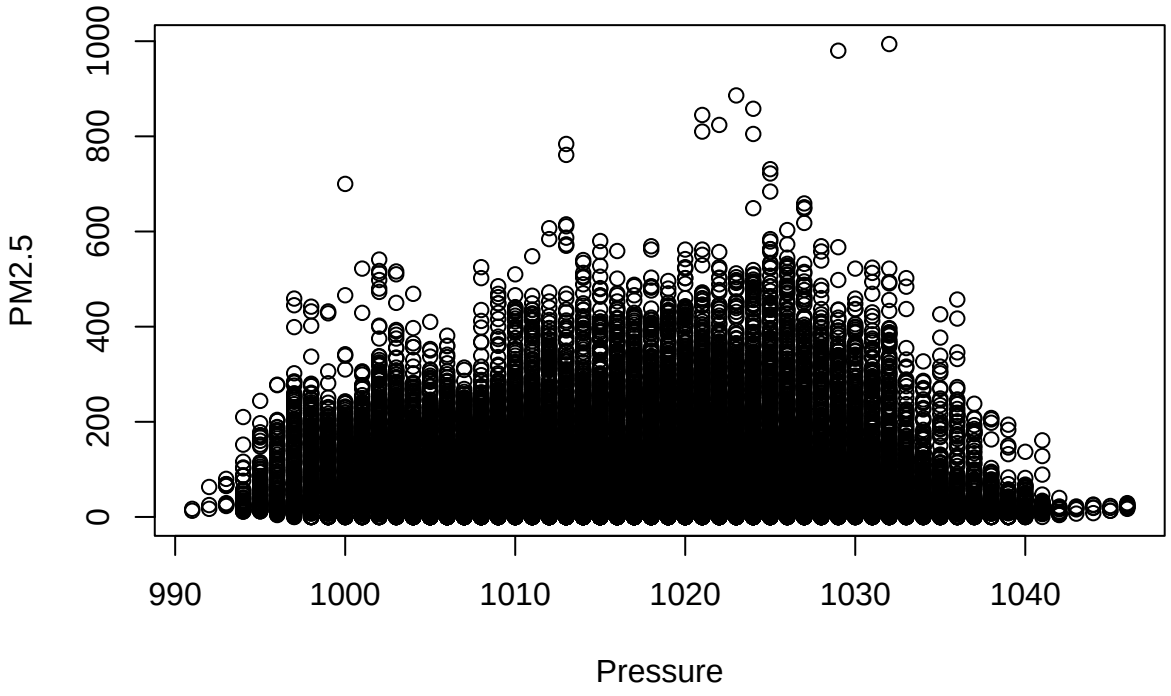
ploting 2 graphs from training data

In this section, I will plot PM2.5- temperature and PM2.5- pressure graph.

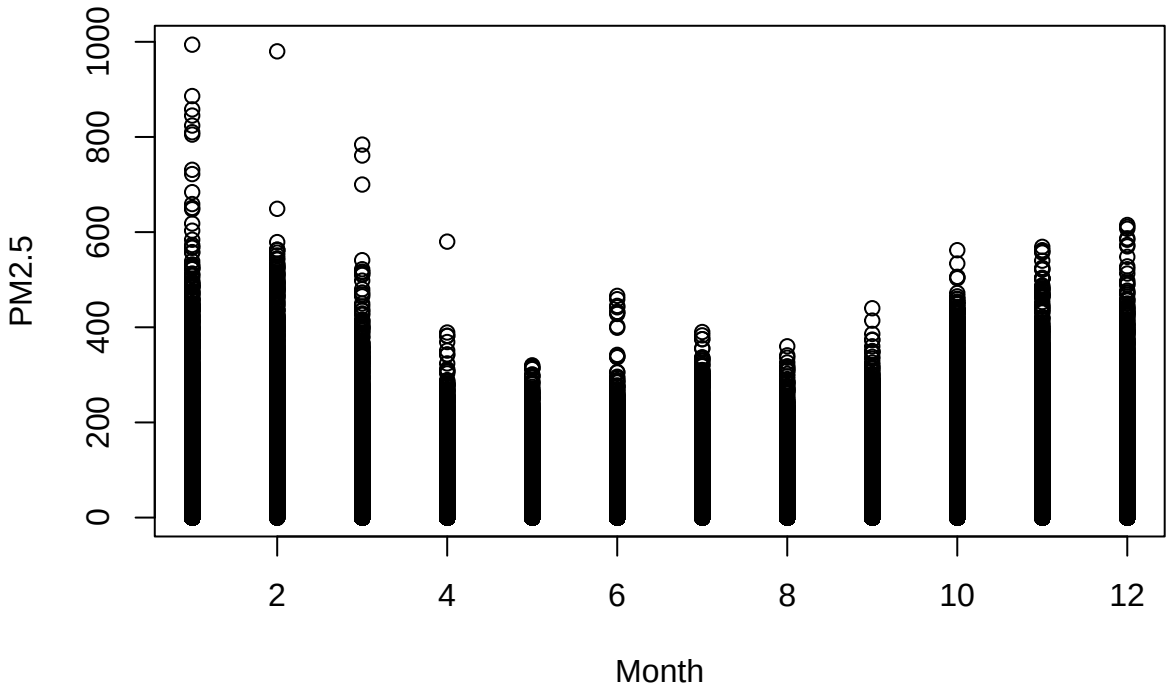
```
plot( train$TEMP,train$pm2.5,
xlab="Temperature", ylab="PM2.5")
```




```
plot( train$PRES,train$pm2.5,
xlab="Pressure", ylab="PM2.5")
```




```
plot( train$month,train$pm2.5,
xlab="Month", ylab="PM2.5")
```



From the graph, we can see that they seems not in linear relation but they are similar to each other.

And the month seems to have some relationship with pm2.5 data

simple linear regression model

Here, I will build a simple linear regression model with temperature as predictor.

```
x<-train$TEMP
y<-train$pm2.5

m1<-lm(y~x)
summary(m1)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -114.92  -68.53  -24.72   40.39  883.20
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 102.56727    0.70214  146.08   <2e-16 ***
## x            -0.68642    0.04025  -17.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 91.81 on 35057 degrees of freedom
## Multiple R-squared:  0.00823,    Adjusted R-squared:  0.008201
## F-statistic: 290.9 on 1 and 35057 DF,  p-value: < 2.2e-16
```

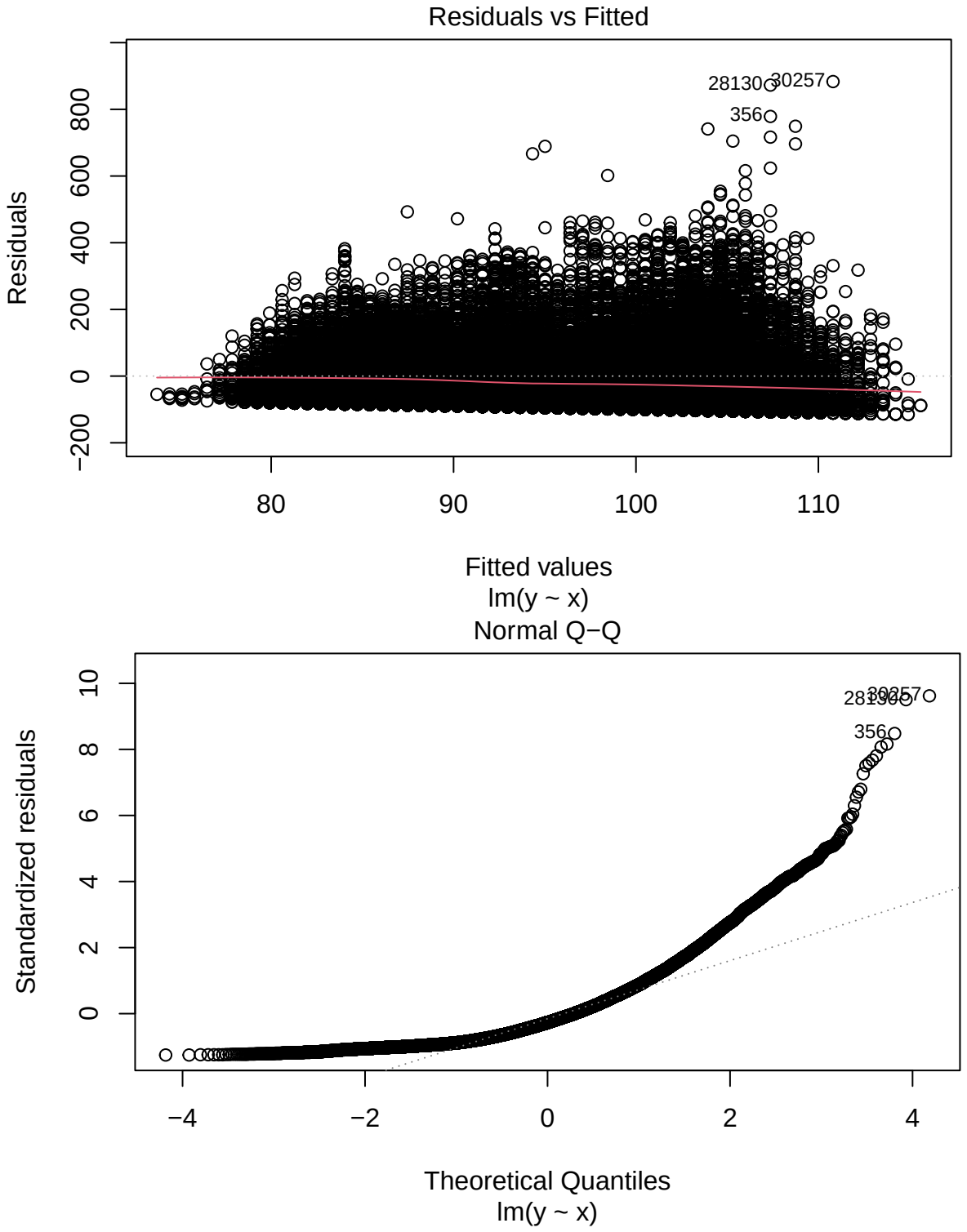
From the summary, degrees of freedom is 91.81 and R-squared is 0.08201 and x estimatet is negative. which

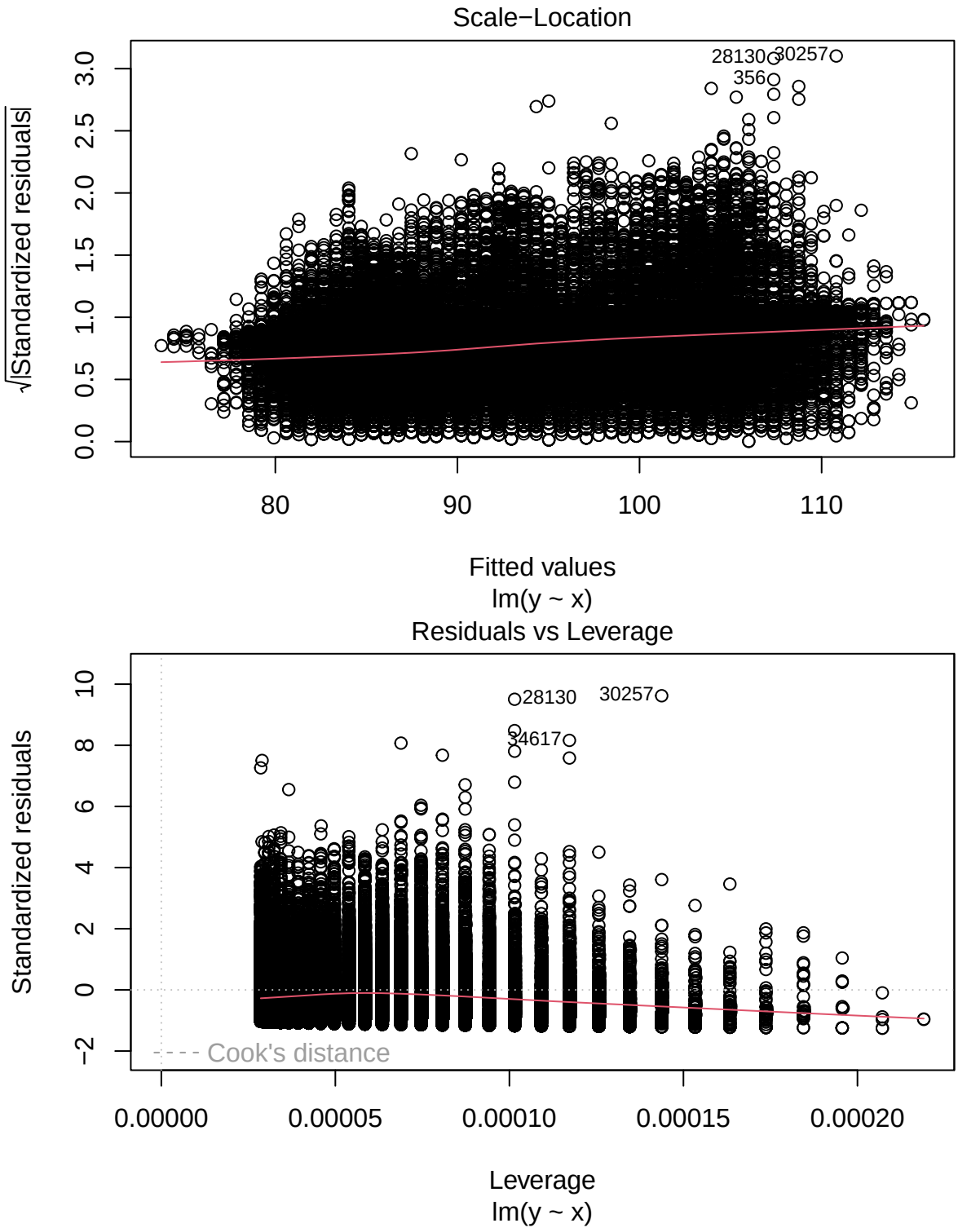
means it doesn't fit to proper linear relation ship.

Therefore, we can say this data cannot be explained with linear relationship of temperature and PM2.5

####ploting residual

The residual is like below.





The first graph shows that all predictions have kind of similar residuals.

The second graph shows that residuals doesn't follow normal distribution.

The third graph shows red line slightly climbing upside in a straight line. This means every value makes a bit similar distribution of residuals.

The fourth graph indicates that residuals has outliers in the low row number.

According to the residual plot, we can say that this model isn't good to explain pm2.5

multi linear regression

from the simple linear regression, I figured out that row number might have some correlation to pm2.5.

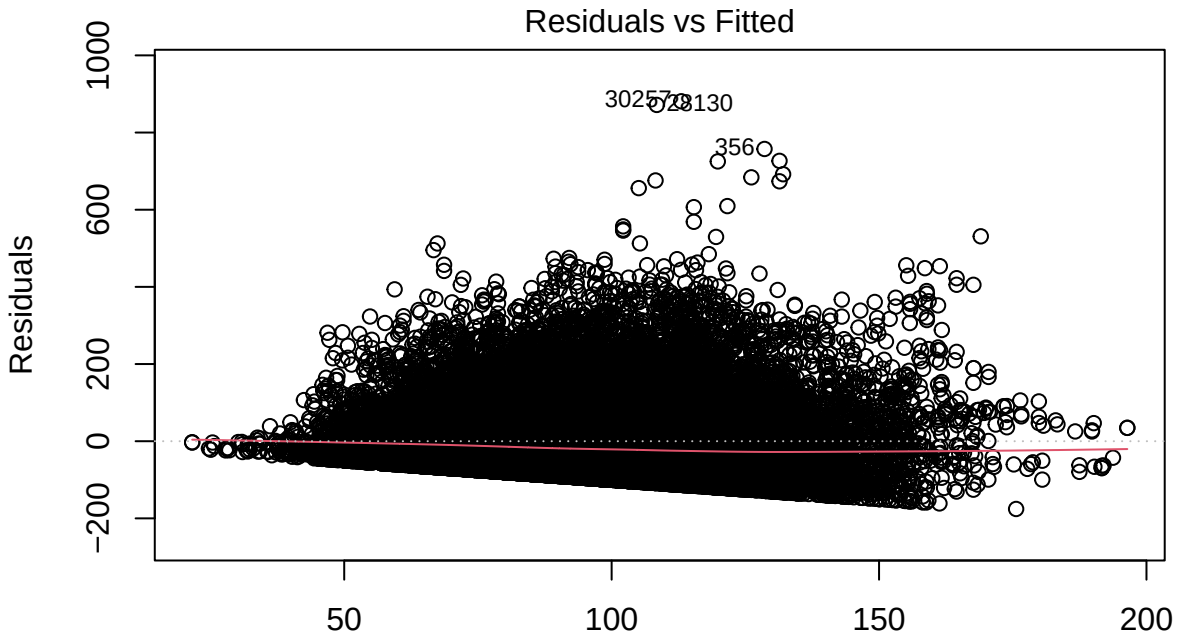
Therefore, i selected month which is related to row number this timeas another predictor.

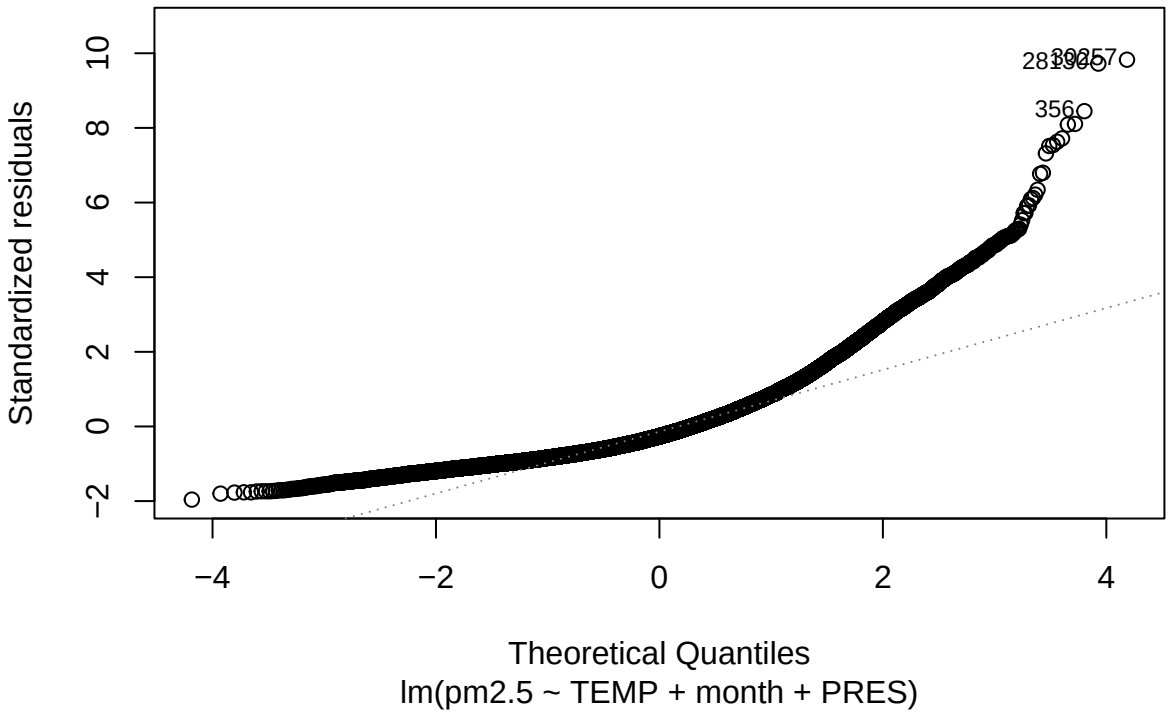
Also, i selected pressure because the plot between pressure and pm2.5 is similar.

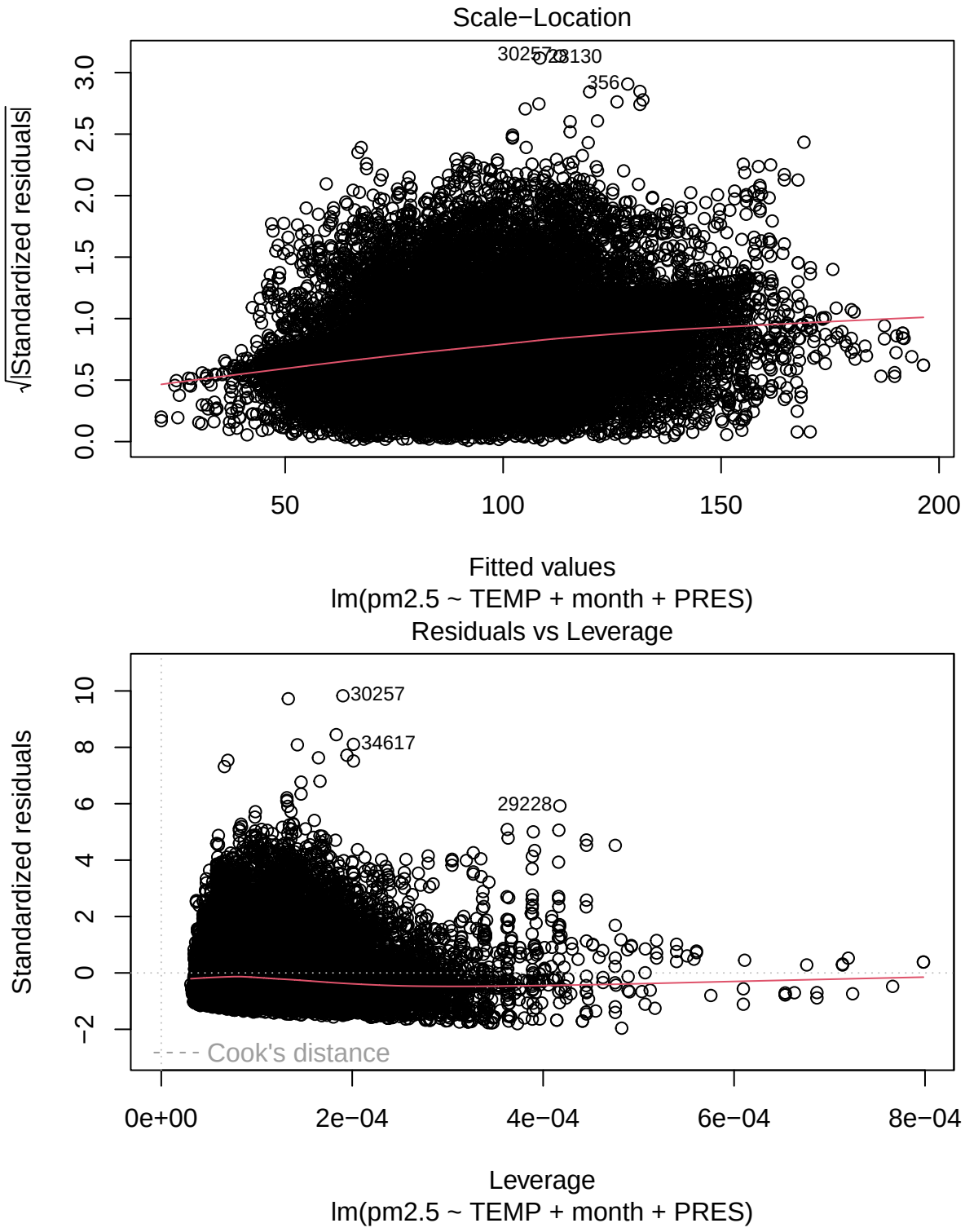
```
m2<-lm(data=train,pm2.5~TEMP+month+PRES)
summary(m2)
```

```
##
## Call:
## lm(formula = pm2.5 ~ TEMP + month + PRES, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -175.62  -62.47  -23.66   37.64  880.98
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3657.35763   85.85207  42.601  < 2e-16 ***
## TEMP          -3.13642    0.07155 -43.835  < 2e-16 ***
## month          0.67931    0.14268   4.761 1.93e-06 ***
## PRES          -3.47156    0.08386 -41.397  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 89.64 on 35055 degrees of freedom
## Multiple R-squared:  0.0545, Adjusted R-squared:  0.05442
## F-statistic: 673.5 on 3 and 35055 DF,  p-value: < 2.2e-16
```

```
plot(m2)
```


Residuals vs Fitted
Residuals
1000
600
200
0
−200
30257
28130
356
50
100
150
200
Fitted values
lm(pm2.5 ~ TEMP + month + PRES)


Normal Q−Q
Standardized residuals
10
8
6
4
2
0
−2
28130
30257
356
−4
−2
0
2
4
Theoretical Quantiles
lm(pm2.5 ~ TEMP + month + PRES)



degrees of freedom lowered, but the linear graph still cannot explain.

Especially, the ploted graphs didn't show any better result than m1.

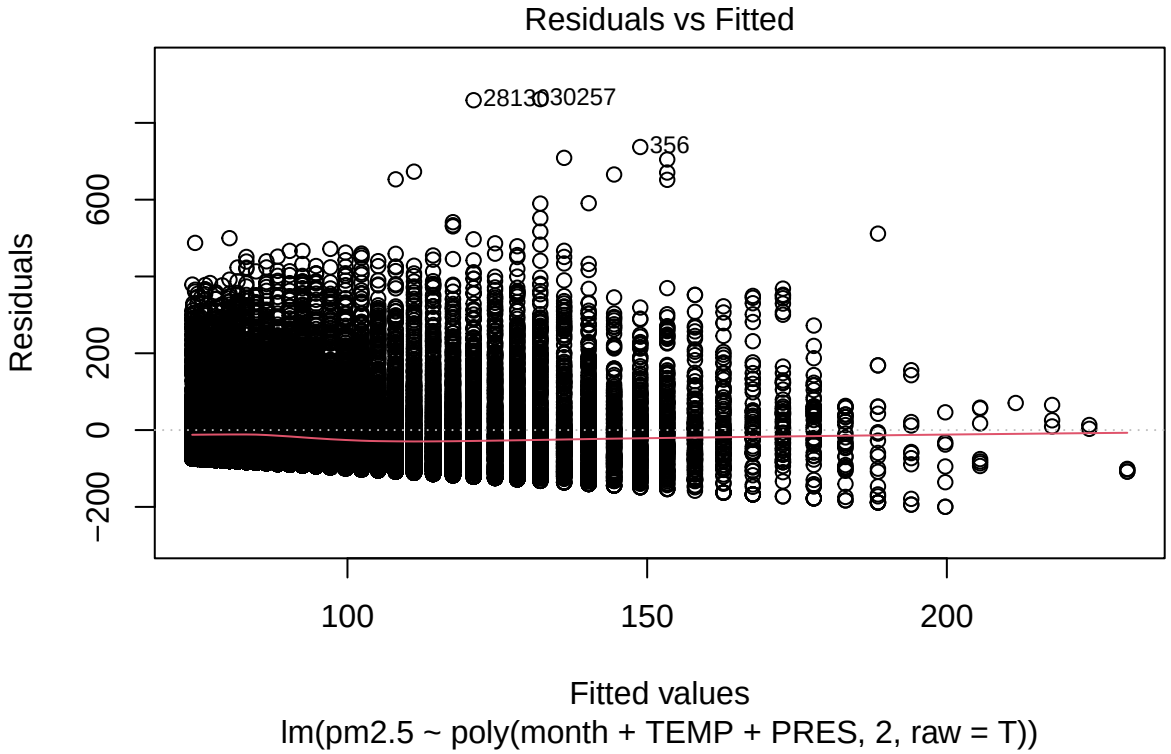
polynomial regression

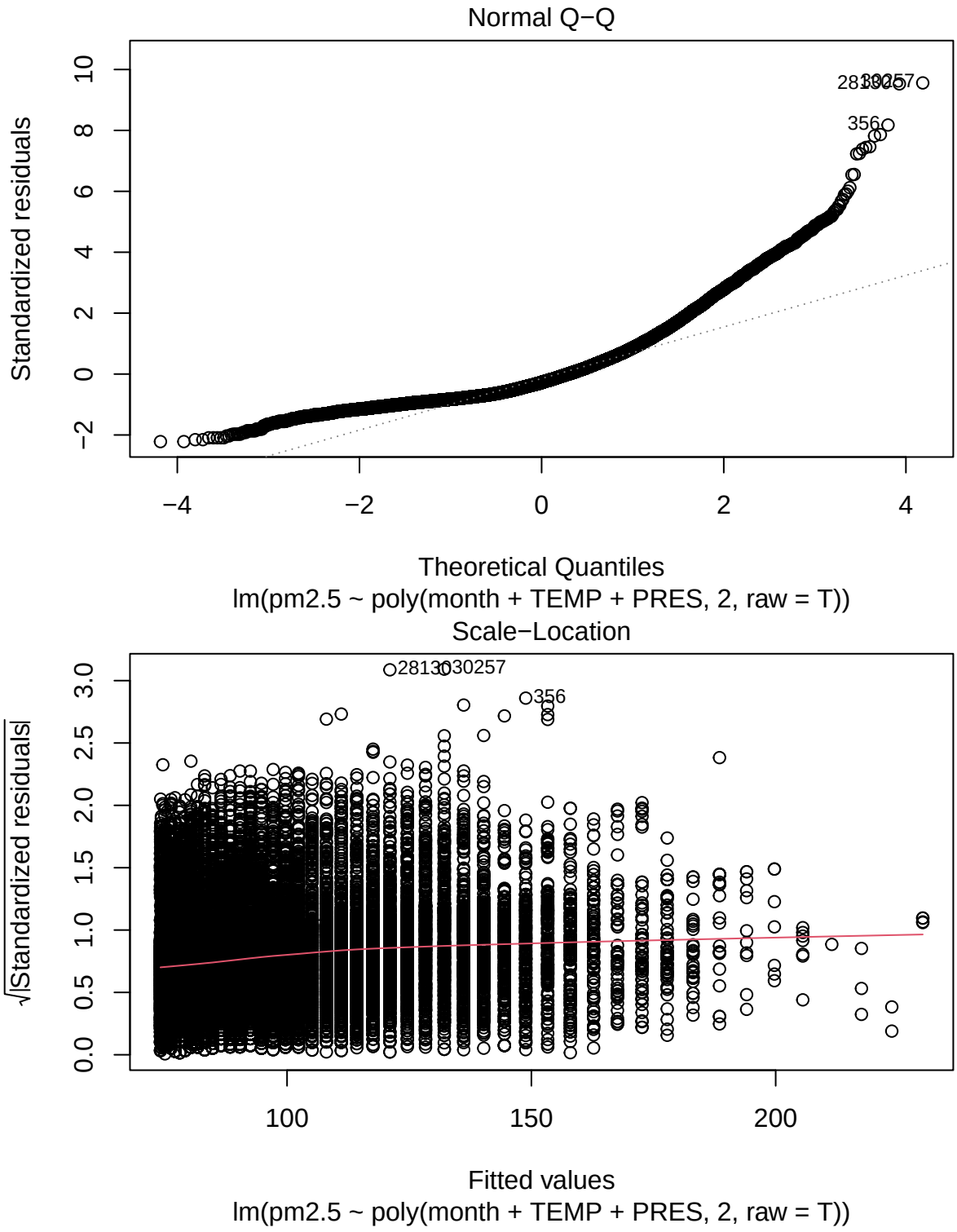
I tried polynomial regression to improve model using month and pm2.5

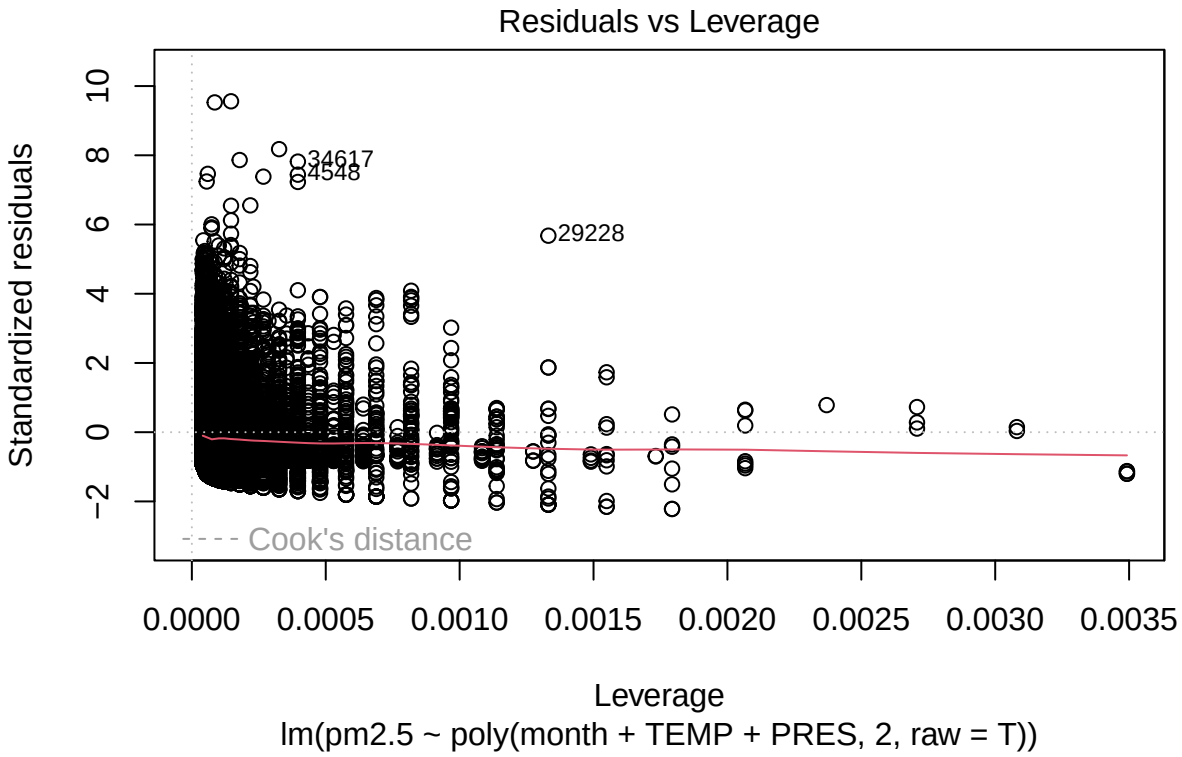
```
m3<-lm(data=train,pm2.5~poly(month+TEMP+PRES,2, raw = T))
summary(m3)
```

```
##
## Call:
## lm(formula = pm2.5 ~ poly(month + TEMP + PRES, 2, raw = T), data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -199.73  -64.66  -24.75   38.34  861.83
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          7.258e+04  5.544e+03   13.09   <2e-16
## poly(month + TEMP + PRES, 2, raw = T)1 -1.380e+02  1.072e+01  -12.87   <2e-16
## poly(month + TEMP + PRES, 2, raw = T)2  6.567e-02  5.185e-03   12.67   <2e-16
##
## (Intercept)                          ***
## poly(month + TEMP + PRES, 2, raw = T)1 ***
## poly(month + TEMP + PRES, 2, raw = T)2 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 90.16 on 35056 degrees of freedom
## Multiple R-squared:  0.04364,    Adjusted R-squared:  0.04359
## F-statistic: 799.9 on 2 and 35056 DF,  p-value: < 2.2e-16
```

```
plot(m3)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(pm2.5 ~ poly(month + TEMP + PRES, 2, raw = T))
Scale−Location
√|Standardized residuals|
Fitted values
lm(pm2.5 ~ poly(month + TEMP + PRES, 2, raw = T))



The sclae-location graph showed some improvement this time. Also, this model best follows normal distribution from the 3 graphs

Again, the modified regression cannot explain the data's correlation.

Comparing results

```
anova(m1,m2,m3)
```

```
## Warning in anova.lmlist(object, ...): models with response 'c("pm2.5", "pm2.5")'
## removed because response differs from model 1
```

```
## Analysis of Variance Table
##
## Response: y
##              Df    Sum Sq Mean Sq F value    Pr(>F)
## x             1   2452005 2452005   290.9 < 2.2e-16 ***
## Residuals 35057 295493389    8429
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I got 3 regression models. Those models all showed difference, but cannot explain the data. All 3 model's estimates are negative and r squared value is moving toward 0.

The first model had the worst normal qq and scale -location result and the third had the best normal qq and scale- location result. However, none of the model showed some correlation to pm2.5

The reason of this situation comes from the data's shape. The data is more like histogram, not a graph.

Test

Test using models

```
print(paste("model1"))
pred1 <- predict(m1, newdata=test)

cor1<-cor(pred1, test$pm2.5)
mse1<-mean((pred1-test$pm2.5)^2)
rmse1<-sqrt(mse1)

print(paste('correlation: ', cor1))
print(paste('mse: ', mse1))
print(paste('rmse: ', rmse1))

print(paste("model2"))
pred1 <- predict(m2, newdata=test)

cor2<-cor(pred2, test$pm2.5)
mse2<-mean((pred2-test$pm2.5)^2)
rmse2<-sqrt(mse2)

print(paste('correlation: ', cor2))
print(paste('mse: ', mse2))
print(paste('rmse: ', rmse2))

print(paste("model3"))
pred1 <- predict(m3, newdata=test)

cor1<-cor(pred3, test$pm2.5)
mse1<-mean((pred3-test$pm2.5)^2)
rmse1<-sqrt(mse3)

print(paste('correlation: ', cor3))
print(paste('mse: ', mse3))
print(paste('rmse: ', rmse3))
```

R code made an error, but the correlation should be low enough to say that all 3 models cannot explain and predict the data properly, since none of 3 models had proper linear relatinoship.

Thus, we can say that temperature, season, and air pressure cannot explain beijing's air condition. There should be other reason causing pm2.5, such as external inflow. This is why 3 models cannot explain the data, because temperature and air pressure is about beijing's condition, not external source's condition.