Regression Analysis for China Air Quality Index and Correlative Factors Dataset

MATH1312_Project

Shiting Yin (s3645072)

1 Introduction

Nowadays, a common issue-haze appear in China especially in winter (J. Liu et al. 2018). The definition of AQI and how AQI works could be seen in (*Air Quality Index (AQI) Basics* 2016). This report based on data “AQI” which recorded information relate to Air Quality Index and correlative factors dataset in China in 2015. The report contain seven sections. This section as first section for introduction. The second section is the part to understand data. Third section will load data into R software doing data preprocessing, tidy and clean data into suitable data format. Then, fourth section will do data exploration, modeling with original and transformed dataset, the hypothesis test will be shown as well. And then, applying best model to predict AQI value for test dataset. Moreover, comparing models and prediction result would be done in sixth part. Finally, based on fitted models and performance on test dataset, the section of summary will be given by this report.

2 Data Set

The Kaggle (https://www.kaggle.com/maxwellnee/china-aqi-test) provides dataset, AQI include 323 observations and 12 attributes, and 319 of this dataset will as training dataset, the last 4 observation will as the test data to valid the predicted value for AQI.
This dataset might collected by observational study, observing AQI factors in each city and then recording these data into data table to insight into them to see the interesting patterns.
The data description show the detail information of response (AQI) and each predictor, the contribution of each predictor could be explored by modeling part. The value of AQI depends on different predictors, such as the Population Density, there is a positive linear relationship between them; the incineration, it seems that this variable relate to GDP with a positive association. So, not just relationship between response and relevant variables, but also some relationship between predictors.
For linear regression, one of the important assumption is that avoid correlation between predictors, how to find and deal with this issue will be show in next section.

2.1 Descriptive Features

The information for this part based on dataset in Kaggle.

2.1.1 Independent variables:

- City: City name
- x1 - Precipitation: Precipitation (mm)
- x2 - GDP: Gross Domestic Product (100 million)
- x3 - Temperature: Temperature (Celsius Degree)
- x4 - Longititude: Longititute (degree)
- x5 - Latitude: Latitude (degree)
- x6 - Altitude: Altitude (metre)
- x7 - PopulationDensity: number of people per square kilometer
- x8 - Coastal: is it Coastal (0-not coastal, 1-coastal)
- x9 - GreenCoverageRate: Rate of Green Coverage in City (%)
- x10 - Incineration: converts the waste into ash, flue gas and heat (10,000ton)

2.1.2 Dependent variable (desired target):

- AQI: Air Quality Index (0-500) (numeric)

3 Data Pre-processing

3.1 Loading required packages and dataset

```
setwd("E:/201901/regression analysis/project/china-aqi-test")
library(car)
library(ggplot2)
library(MASS)
library(formattable)
library(dplyr)
library(forecast)
library(magicfor)
data = read.csv('CompletedDataset.csv')[1:319,]
#head(data,3)
```

Applying regression for this dataset, it seems that the city column as index for each row, so, considering remove this attribute.

```
data_no_city = data[-1]
head(data_no_city,3)
```

```
##   AQI Precipitation    GDP Temperature Longititute Latitude Altitude
## 1  23         665.1 271.13     8.20000    102.22465 31.89941     2617
## 2 137          80.4 610.00    12.27671     80.26338 41.16754     1108
## 3  85         150.0 322.58    24.20000    105.72895 38.85192     1673
##   PopulationDensity Coastal GreenCoverageRate Incineration.10.000ton.
## 1                11       0             36.00                      23
## 2              6547       0             33.94                      23
## 3                 1       0             36.00                      23
```

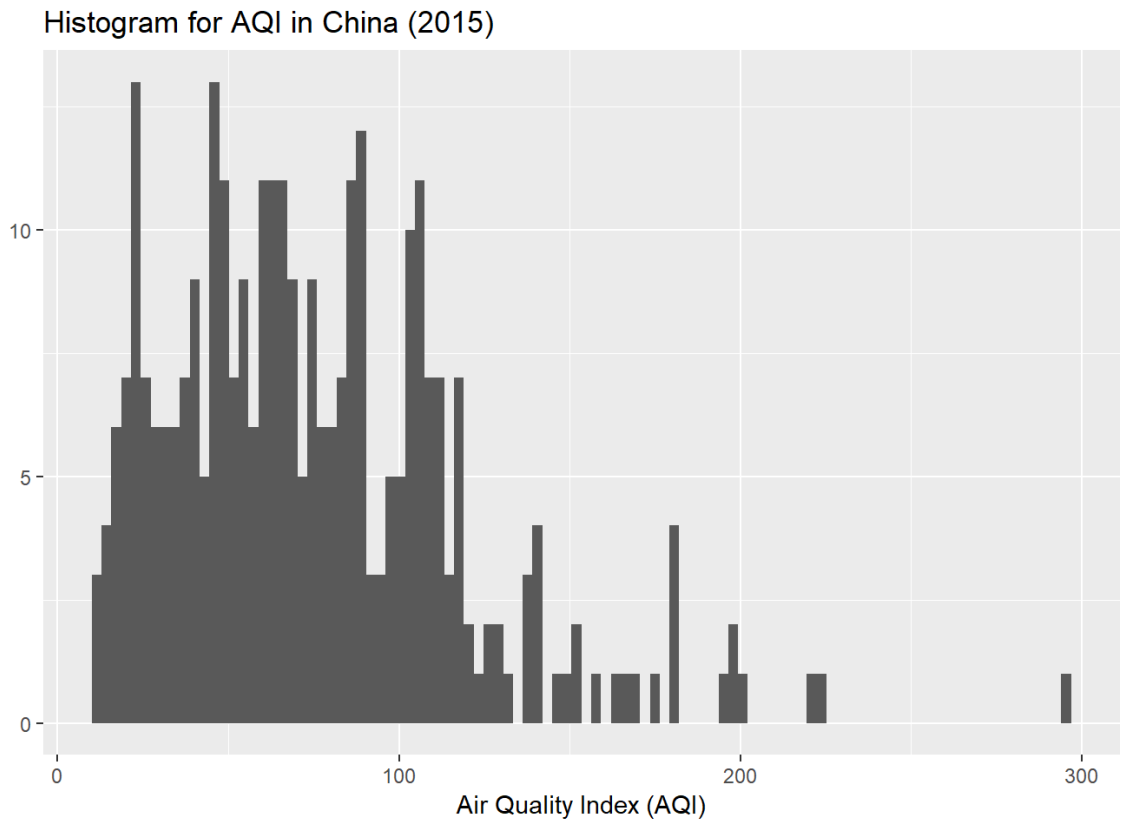
```
colSums(is.na(data_no_city))
```

```
##                     AQI           Precipitation                     GDP
##                       0                       0                       0
##             Temperature             Longititute                Latitude
##                       0                       0                       0
##                Altitude       PopulationDensity                 Coastal
##                       0                       0                       0
##       GreenCoverageRate Incineration.10.000ton.
##                       0                       0
```

Before doing further processing, considering whether there are some missing values in this dataset, if there are some missing value, this issue should be done at this stage with reasonable method. Fortunately, there is not any missing value in this data.

3.2 Dependent Attribute (AQI)

```
ggplot(data = data, aes(x = AQI)) + geom_histogram(bins = 100) +
  labs(title = "Histogram for AQI in China (2015)",
       x = "Air Quality Index (AQI)", y = "")
```



The figure shows that the histogram of dependent variable AQI, it seems that most of values between 25 to 150, just few of them above 200, say, air quality is not quite bad in 2015.

4 Data Exploration

For this project, applying regression method to fit model, the correlated variables should be avoid. In order to see the correlation (linear) between independent variables, different methods could be applied in this part. In fact, the high correlated variable could be replaced by one of them, because others could be get by one of them, and remove correlated attribute means that avoid multicollinearity in regression models.

4.1 Correlation

4.1.1 Giving alias to each attribute

```
str(data_no_city)
```

```
## 'data.frame':    319 obs. of  11 variables:
##  $ AQI                     : int  23 137 85 28 79 110 111 44 53 58 ...
##  $ Precipitation           : num  665.1 80.4 150 74.2 2127.8 ...
##  $ GDP                     : num  271.1 610 322.6 37.4 1613.2 ...
##  $ Temperature             : num  8.2 12.3 24.2 1 17.3 ...
##  $ Longititute             : num  102.2 80.3 105.7 80.1 117 ...
##  $ Latitude                : num  31.9 41.2 38.9 32.5 30.5 ...
##  $ Altitude                : num  2617 1108 1673 4280 13 ...
##  $ PopulationDensity       : int  11 6547 1 1 2271 4735 2534 20547 5093 2653 ...
##  $ Coastal                 : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ GreenCoverageRate       : num  36 33.9 36 36 45.8 ...
##  $ Incineration.10.000ton.: num  23 23 23 23 27.5 ...
```

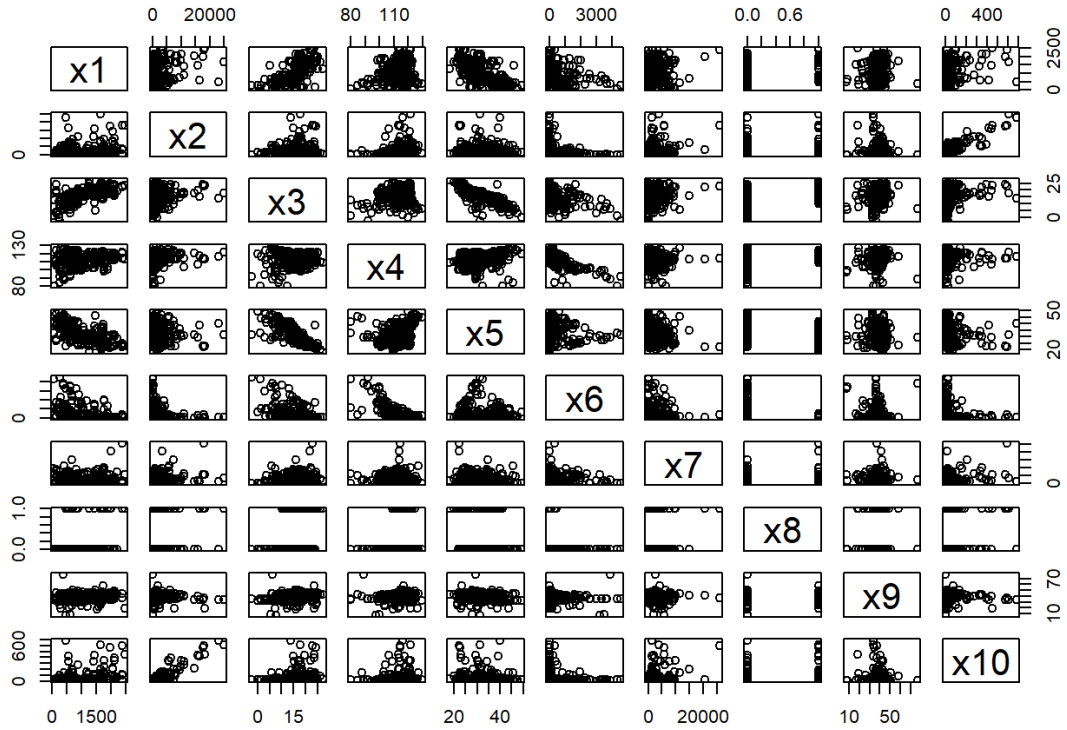
```
# change names of column into x and y
colnames(data_no_city) = c('y','x1','x2','x3','x4','x5','x6','x7','x8','x9','x10')
# assign all predictors to X
X = data_no_city[-1]
head(X,3)
```

```
##       x1      x2       x3        x4       x5   x6   x7 x8    x9 x10
## 1 665.1 271.13  8.20000 102.22465 31.89941 2617   11  0 36.00  23
## 2  80.4 610.00 12.27671  80.26338 41.16754 1108 6547  0 33.94  23
## 3 150.0 322.58 24.20000 105.72895 38.85192 1673    1  0 36.00  23
```

```
# function for values
sign_formatter = formatter("span",
           style = x~style(color=ifelse(abs(x)>0.8 & abs(x)<1.0, 'red',
                                        ifelse(abs(x)<0.5|abs(x)==1.0, 'black','vermilion')))
  )
```

4.1.2 Correlation between independent attributes

```
### correlation between X
# plot for correlation
pairs(X)
```




```
# calculate value for two variables and apply function
cor_x = as.data.frame(cor(X))
sign_x = formattable(cor_x, list(x1=sign_formatter,
                                 x2=sign_formatter,
                                 x3=sign_formatter,
                                 x4=sign_formatter,
                                 x5=sign_formatter,
                                 x6=sign_formatter,
                                 x7=sign_formatter,
                                 x8=sign_formatter,
                                 x9=sign_formatter,
                                 x10=sign_formatter))
#sign_x
sign_x %>% knitr::kable(caption = 'Correlation between Dependent Variables')
```

Correlation between Dependent Variables

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	1.0000000	0.1790559	0.6877373	0.2243499	-0.6575112	-0.3267882	0.0632795	0.2572567	0.1529383	0.2003550

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x2	0.1790559	1.0000000	0.1474059	0.1704663	-0.0124336	-0.2085721	0.2302304	0.1742738	-0.0381060	0.9004213
x3	0.6877373	0.1474059	1.0000000	0.1460550	-0.8075443	-0.4609114	0.1452615	0.3079676	0.2156128	0.1741809
x4	0.2243499	0.1704663	0.1460550	1.0000000	0.1691601	-0.7407649	-0.1233068	0.3729679	0.1621459	0.0702609
x5	-0.6575112	-0.0124336	-0.8075443	0.1691601	1.0000000	0.0049874	-0.1663350	-0.2060642	-0.1400454	-0.0820071
x6	-0.3267882	-0.2085721	-0.4609114	-0.7407649	0.0049874	1.0000000	-0.0331784	-0.2722789	-0.1851984	-0.1223851
x7	0.0632795	0.2302304	0.1452615	-0.1233068	-0.1663350	-0.0331784	1.0000000	-0.0359282	0.0199455	0.2832129
x8	0.2572567	0.1742738	0.3079676	0.3729679	-0.2060642	-0.2722789	-0.0359282	1.0000000	0.2669368	0.1574701
x9	0.1529383	-0.0381060	0.2156128	0.1621459	-0.1400454	-0.1851984	0.0199455	0.2669368	1.0000000	-0.0286509
x10	0.2003550	0.9004213	0.1741809	0.0702609	-0.0820071	-0.1223851	0.2832129	0.1574701	-0.0286509	1.0000000

Based on plot, it seems that the 10*10 matrix could not give a clear picture on it. The table using value to show the correlation between each two attribute is much better than plot. Based on table, it shows that the x2 and x10 have high correlation which is 0.89955027, x3 and x5 have high correlation which is 0.8071193. There are other attributes have high correlation as well, which could be seen in table.

According to these information, x2 and x10, x3 and x5 have high linear correlation, include all of them into regression models might lead to multicollinearity, considering remove one of them from each pair. Trying fit regression model with all of predictors, and then to see the vif values for them further.

4.2 Modeling

4.2.1 Fitting model with original data

In this part, applying different fitted models with different predictors on original data, then test regression assumption (both plot and statistic) on fitted models, the plot could give whole picture on each fitted model while statistic could give objective test result on it. If there are some assumptions are violated, trying transformation on dataset, then to see whether the performance on the fitted model getting better or not.

4.2.1.1 Model with all of predictors

```
# regression for full model
full = lm(y~., data = data_no_city)
null = lm(y~1, data = data_no_city)
vif(full)
```

```
##       x1       x2       x3       x4       x5       x6       x7       x8
## 2.367498 5.767398 8.013020 3.137639 6.777899 4.936335 1.181640 1.380922
##       x9      x10
## 1.123211 5.797954
```

```
summary(full)  ###Adjusted R-squared:  0.4381
```

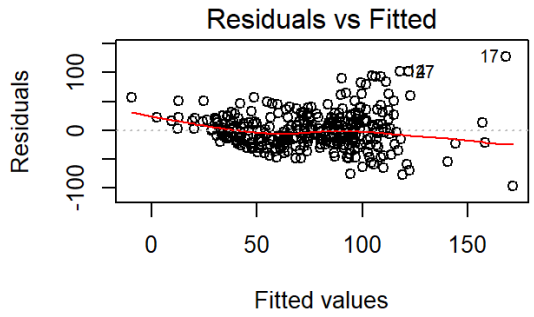
```
##
## Call:
## lm(formula = y ~ ., data = data_no_city)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -97.297 -19.120  -3.415  16.681 127.820
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.6407401 67.4450741   0.588  0.55713
## x1          -0.0155411  0.0047571  -3.267  0.00121 **
## x2           0.0013896  0.0013267   1.047  0.29573
## x3           3.0815651  1.0149309   3.036  0.00260 **
## x4          -1.3100712  0.4164184  -3.146  0.00182 **
## x5           5.1629261  0.7706162   6.700 9.93e-11 ***
## x6          -0.0160350  0.0053911  -2.974  0.00317 **
## x7          -0.0002673  0.0006728  -0.397  0.69148
## x8          -6.0270241  4.9070156  -1.228  0.22029
## x9          -0.2132142  0.3025039  -0.705  0.48145
## x10          0.0229407  0.0471839   0.486  0.62717
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.33 on 308 degrees of freedom
## Multiple R-squared:  0.4557, Adjusted R-squared:  0.4381
## F-statistic: 25.79 on 10 and 308 DF,  p-value: < 2.2e-16
```

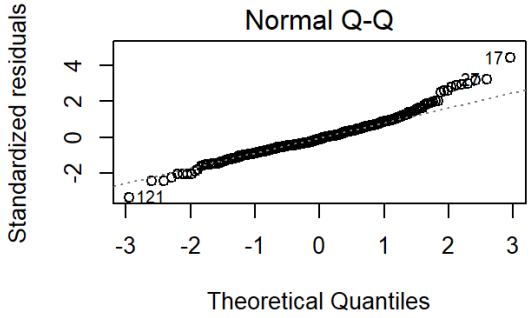
```
anova(null,full)
```

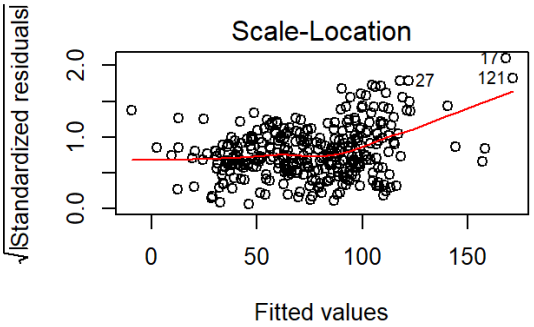
```
## Analysis of Variance Table
##
## Model 1: y ~ 1
## Model 2: y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    318 591418
## 2    308 321894 10    269523 25.789 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

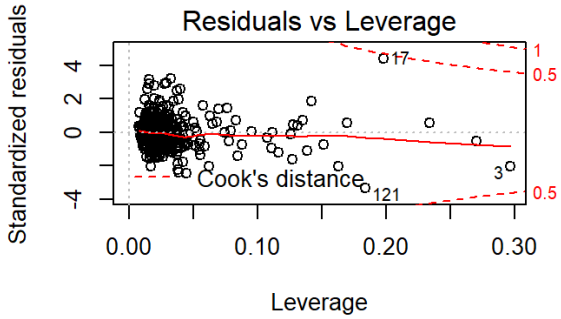
It shows that the vif values of x2, x3, x5 and x10 are greater than 5, it means that same result as we got before. Based on summary and Anova information, it shows that only 45.57% of variances in the dataset could be explained by estimated model. And the sum of square residuals 322861 on 312 degrees of freedom, sum of regression is 272399 on 10 degrees of freedom. This model, x1, x3, x4, x5 and x6 are significant at the 5% level of significance.

```
par(mfrow=c(2,2))
plot(full)
```










```
# stat test
ncvTest(full)   ### p = < 2.22e-16
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 74.93081, Df = 1, p = < 2.22e-16
```

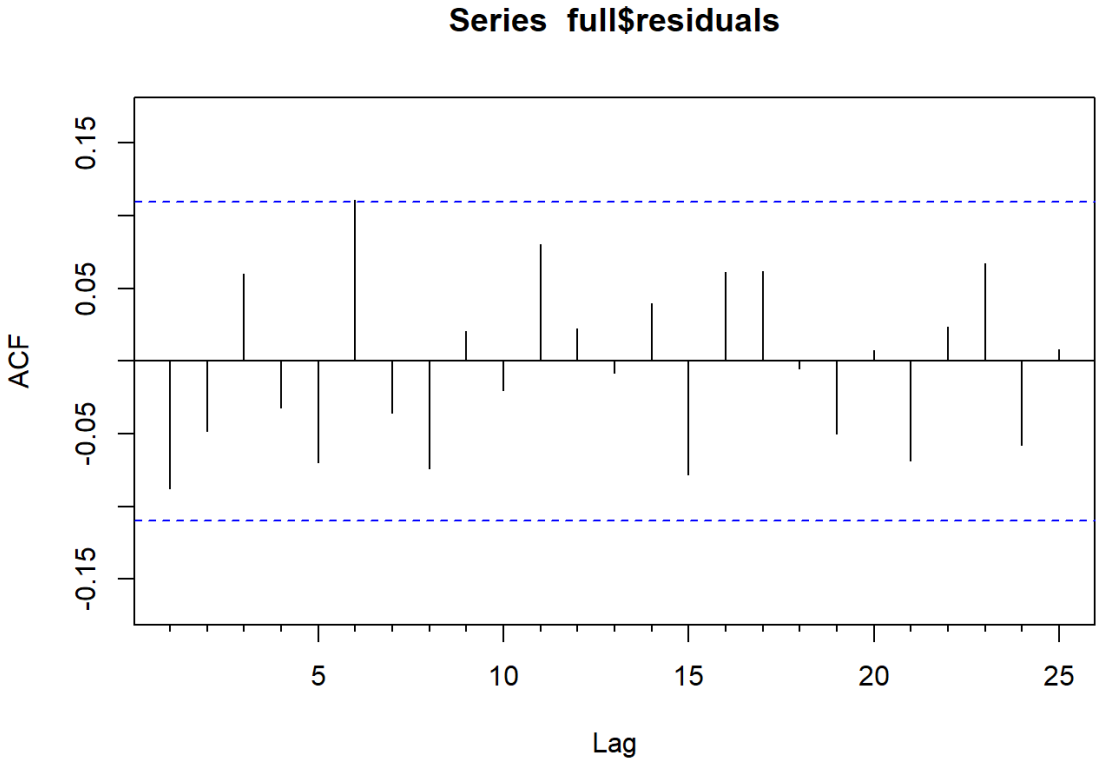
```
# 1st lag auto-correlation
durbinWatsonTest(full)    ###0.106
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.08788075      2.175178   0.118
##  Alternative hypothesis: rho != 0
```

```
shapiro.test(full$residuals)    ###2.534e-06
```

```
##
##  Shapiro-Wilk normality test
##
## data:  full$residuals
## W = 0.96914, p-value = 2.534e-06
```

```
par(mfrow=c(1,1))
Acf(full$residuals)
```



The plot shows that all of assumptions for residuals are violated. Then, the statistic shows that there is not auto-correlation in residuals at first lag while sixth lag has auto-correlation. The constant variance, normally distributed are violated for this case.

4.2.1.2 Deal with multicollinearity

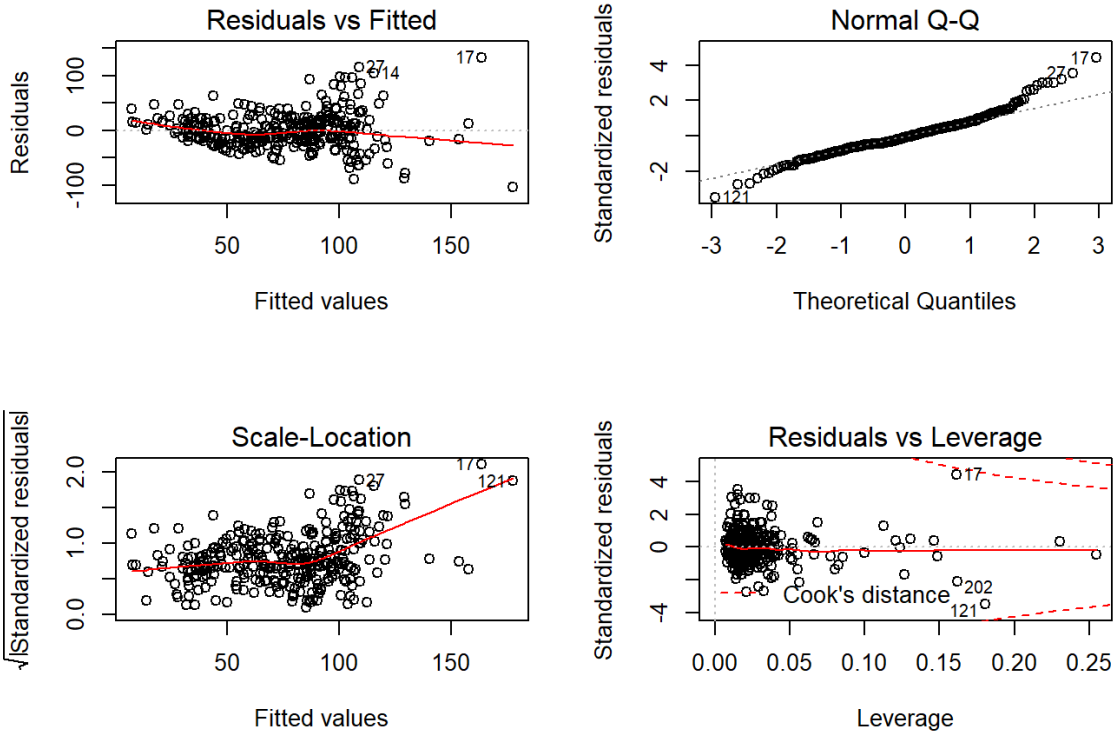
```
# deal with multicollinearity remove x3, x10
reg_no_x3_x10 = lm(y~.-x3-x10, data = data_no_city)
vif(reg_no_x3_x10)
```

```
##       x1       x2       x4       x5       x6       x7       x8       x9
## 2.342875 1.181183 2.875301 2.386489 2.464833 1.145502 1.361958 1.118951
```

```
summary(reg_no_x3_x10)  ###Adjusted R-squared:  0.4241
```

```
##
## Call:
## lm(formula = y ~ . - x3 - x10, data = data_no_city)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.352  -17.659   -2.954   16.685  132.607
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.918e+02  4.628e+01   4.144 4.41e-05 ***
## x1          -1.433e-02  4.791e-03  -2.992 0.002996 **
## x2           2.187e-03  6.078e-04   3.598 0.000374 ***
## x4          -1.685e+00  4.036e-01  -4.176 3.86e-05 ***
## x5           3.274e+00  4.629e-01   7.072 1.02e-11 ***
## x6          -2.739e-02  3.857e-03  -7.102 8.47e-12 ***
## x7          -3.783e-04  6.707e-04  -0.564 0.573139
## x8          -4.254e+00  4.934e+00  -0.862 0.389208
## x9          -1.551e-01  3.057e-01  -0.508 0.612135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.73 on 310 degrees of freedom
## Multiple R-squared:  0.4385, Adjusted R-squared:  0.4241
## F-statistic: 30.27 on 8 and 310 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(reg_no_x3_x10)
```




```
# stat test
ncvTest(reg_no_x3_x10)   ### p = < 2.22e-16
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 94.5141, Df = 1, p = < 2.22e-16
```

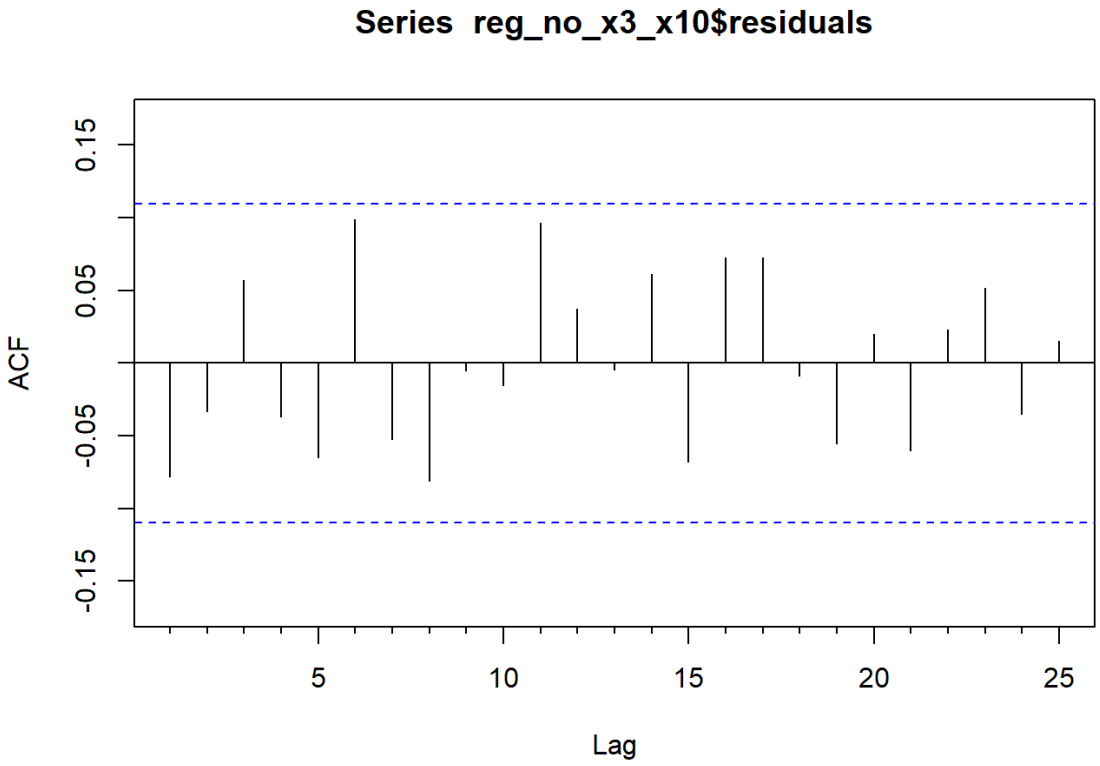
```
shapiro.test(reg_no_x3_x10$residuals)    ###2.221e-07
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reg_no_x3_x10$residuals
## W = 0.96205, p-value = 2.221e-07
```

```
durbinWatsonTest(reg_no_x3_x10)    ###0.168
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      -0.07785934       2.155032    0.19
##  Alternative hypothesis: rho != 0
```

```
par(mfrow=c(1,1))
Acf(reg_no_x3_x10$residuals)
```



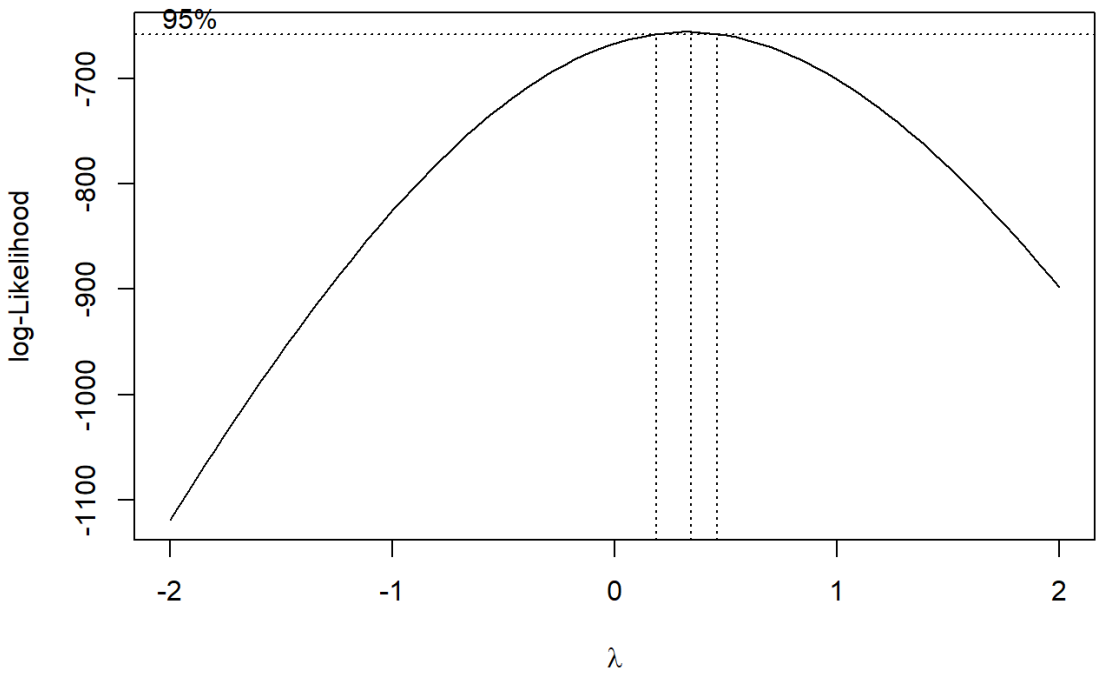
According to these result, it shows that there is no multicollinearity in this dataset because all of vif values are less than 5.
The summary information shows that 43.85% of variances in the dataset could be explained by estimated model. This regression model, x1,x2,x4,x5,x6 are significant at the 5% level of significance while x7, x8 and x9 are not significant.
The residuals test shows that the residuals around zero mean level, and there is not auto-correlation between residuals, but the constant variance and normally distributed are violated.

4.2.2 Fitting model with transformed data

4.2.2.1 Deal with non-constant variance

Based on models fitted before, it shows that one big issue for residuals is non-constant variance, considering transformation (BoxCox) on original data.

```
# deal with multicollinearity remove x3, x10
bc = boxcox(full)
```




```
trans = bc$x[which.max(bc$y)]
lambda = round(trans,1)
lambda       ###0.3
```

```
## [1] 0.3
```

The boxcox() function gives the best result for lambda is 0.3. Then, applying these lambda on dependent variable to create a new variable.

```
# create y_bc with suitable lambda
data_bc = data_no_city
data_bc$y_bc = (data_bc$y ^ lambda - 1)/lambda
data_bc = data_bc[-1]
head(data_bc,3)
```

```
##       x1     x2       x3        x4       x5   x6   x7 x8    x9 x10
## 1 665.1 271.13  8.20000 102.22465 31.89941 2617   11  0 36.00  23
## 2  80.4 610.00 12.27671  80.26338 41.16754 1108 6547  0 33.94  23
## 3 150.0 322.58 24.20000 105.72895 38.85192 1673    1  0 36.00  23
##        y_bc
## 1  5.205472
## 2 11.251280
## 3  9.305425
```

```
reg_trans = lm(y_bc~., data = data_bc)
vif(reg_trans)
```

```
##       x1       x2       x3       x4       x5       x6       x7       x8
## 2.367498 5.767398 8.013020 3.137639 6.777899 4.936335 1.181640 1.380922
##       x9      x10
## 1.123211 5.797954
```

```
summary(reg_trans)   ###0.4643
```

```
##
## Call:
## lm(formula = y_bc ~ ., data = data_bc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9734 -0.9413  0.1020  0.9968  3.2567
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.095e+00  3.220e+00   2.514 0.012435 *
## x1          -6.248e-04  2.271e-04  -2.751 0.006289 **
## x2           7.034e-05  6.333e-05   1.111 0.267569
## x3           1.116e-01  4.845e-02   2.303 0.021952 *
## x4          -7.080e-02  1.988e-02  -3.561 0.000427 ***
## x5           2.450e-01  3.679e-02   6.660 1.26e-10 ***
## x6          -9.805e-04  2.574e-04  -3.810 0.000168 ***
## x7           7.595e-06  3.212e-05   0.236 0.813222
## x8          -4.408e-01  2.342e-01  -1.882 0.060825 .
## x9          -5.609e-03  1.444e-02  -0.388 0.697997
## x10          6.812e-05  2.252e-03   0.030 0.975894
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.543 on 308 degrees of freedom
## Multiple R-squared:  0.4811, Adjusted R-squared:  0.4643
## F-statistic: 28.56 on 10 and 308 DF,  p-value: < 2.2e-16
```

According to these result, it shows that there is multicollinearity in transformed dataset, the vif values of x2, x3, x5 and x10 are greater than 5. The summary information shows that 48.11% of variances in the dataset could be explained by estimated model which better than before. This regression model, x1,x3,x4,x5,x6 are significant at the 5% level of significance.
Then, trying step function to get the selected variables, and then fit a model with these selected predictors.

```
step(reg_trans, direction = 'both')
```

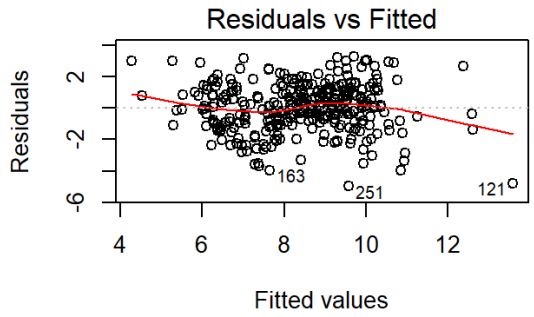
```
## Start:  AIC=287.63
## y_bc ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10
##
##        Df Sum of Sq    RSS    AIC
## - x10   1     0.002 733.56 285.64
## - x7    1     0.133 733.69 285.69
## - x9    1     0.359 733.91 285.79
## - x2    1     2.938 736.49 286.91
## <none>              733.55 287.63
## - x8    1     8.433 741.99 289.28
## - x3    1    12.630 746.18 291.08
## - x1    1    18.028 751.58 293.38
## - x4    1    30.209 763.76 298.51
## - x6    1    34.569 768.12 300.32
## - x5    1   105.635 839.19 328.55
##
## Step:  AIC=285.64
## y_bc ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9
##
##        Df Sum of Sq    RSS    AIC
## - x7    1     0.142 733.70 283.70
## - x9    1     0.359 733.92 283.79
## <none>              733.56 285.64
## - x8    1     8.434 741.99 287.28
## + x10   1     0.002 733.55 287.63
## - x3    1    12.723 746.28 289.12
## - x2    1    14.834 748.39 290.02
## - x1    1    18.096 751.65 291.41
## - x4    1    30.332 763.89 296.56
## - x6    1    34.884 768.44 298.46
## - x5    1   105.927 839.48 326.66
##
## Step:  AIC=283.7
## y_bc ~ x1 + x2 + x3 + x4 + x5 + x6 + x8 + x9
##
##        Df Sum of Sq    RSS    AIC
## - x9    1     0.340 734.04 281.85
## <none>              733.70 283.70
## - x8    1     8.585 742.28 285.41
## + x7    1     0.142 733.56 285.64
## + x10   1     0.011 733.69 285.69
## - x3    1    12.590 746.29 287.12
## - x2    1    16.784 750.48 288.91
## - x1    1    18.488 752.19 289.64
## - x4    1    31.871 765.57 295.26
## - x6    1    36.450 770.15 297.16
## - x5    1   107.230 840.93 325.21
##
## Step:  AIC=281.85
## y_bc ~ x1 + x2 + x3 + x4 + x5 + x6 + x8
##
##        Df Sum of Sq    RSS    AIC
## <none>              734.04 281.85
## + x9    1     0.340 733.70 283.70
## + x7    1     0.123 733.92 283.79
## + x10   1     0.009 734.03 283.84
## - x8    1     9.634 743.67 284.00
## - x3    1    12.395 746.43 285.19
## - x2    1    17.578 751.62 287.39
## - x1    1    18.496 752.53 287.78
## - x4    1    32.025 766.06 293.47
## - x6    1    36.291 770.33 295.24
## - x5    1   107.129 841.17 323.30
```

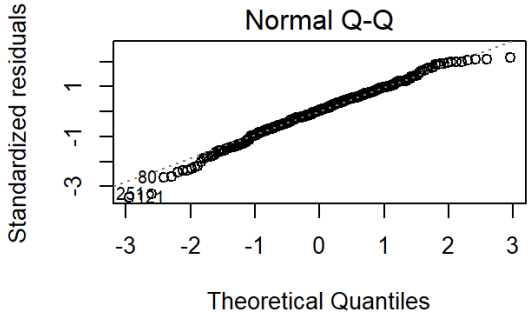
```
##
## Call:
## lm(formula = y_bc ~ x1 + x2 + x3 + x4 + x5 + x6 + x8, data = data_bc)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4
##   8.092e+00   -6.290e-04    7.513e-05    1.097e-01   -7.178e-02
##          x5           x6           x8
##   2.436e-01   -9.869e-04   -4.610e-01
```

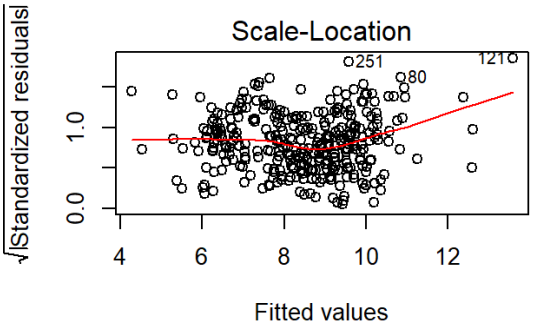
```
reg_trans1 = lm(y_bc~x1+x2+x3+x4+x5+x6+x8, data = data_bc)
summary(reg_trans1)
```

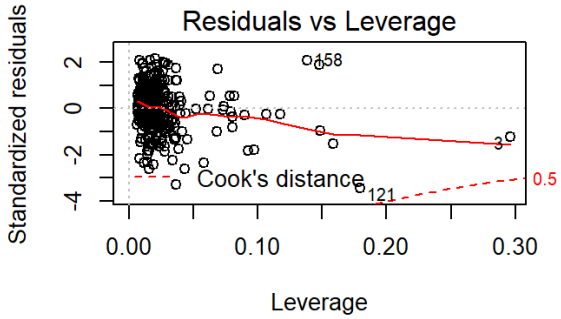
```
##
## Call:
## lm(formula = y_bc ~ x1 + x2 + x3 + x4 + x5 + x6 + x8, data = data_bc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9709 -0.9279  0.0771  1.0012  3.2769
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.092e+00  3.098e+00   2.612 0.009443 **
## x1          -6.290e-04  2.247e-04  -2.799 0.005440 **
## x2           7.513e-05  2.753e-05   2.729 0.006714 **
## x3           1.097e-01  4.787e-02   2.292 0.022596 *
## x4          -7.178e-02  1.949e-02  -3.684 0.000271 ***
## x5           2.436e-01  3.615e-02   6.737 7.83e-11 ***
## x6          -9.869e-04  2.517e-04  -3.921 0.000108 ***
## x8          -4.610e-01  2.282e-01  -2.020 0.044203 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.536 on 311 degrees of freedom
## Multiple R-squared:  0.4808, Adjusted R-squared:  0.4691
## F-statistic: 41.14 on 7 and 311 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(reg_trans1)
```










```
# stat test
ncvTest(reg_trans1)  ### 0.041317
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 4.162953, Df = 1, p = 0.041317
```

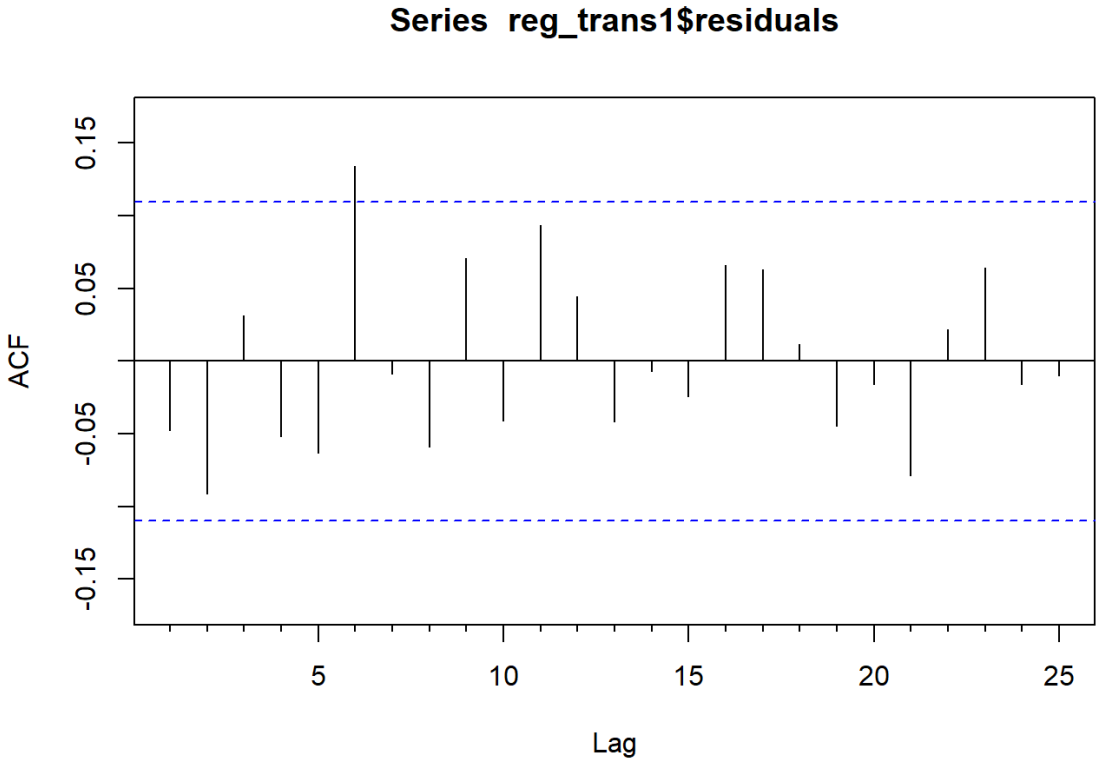
```
shapiro.test(reg_trans1$residuals)   ###0.02521
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reg_trans1$residuals
## W = 0.9898, p-value = 0.02521
```

```
durbinWatsonTest(reg_trans1)   ###0.412
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.04773085       2.09334   0.368
##  Alternative hypothesis: rho != 0
```

```
par(mfrow=c(1,1))
Acf(reg_trans1$residuals)
```



The residuals test shows that the residuals around zero mean level until point 11 for fitted value. The residuals and leverage plot shows that the observation 121 and 158 outside of red dash line with high residual and high leverage, this might be the evidence for outlier. There is no auto-correlation at first lag while appear at sixth lag. The constant variance and normally distributed are violated as well for transformed dataset.

4.2.2.2 Model without outlier 121 and 158

Based on models fitted before, it shows that some of outliers (121 and 158) in the dataset, trying remove them to fit model to see the performance.

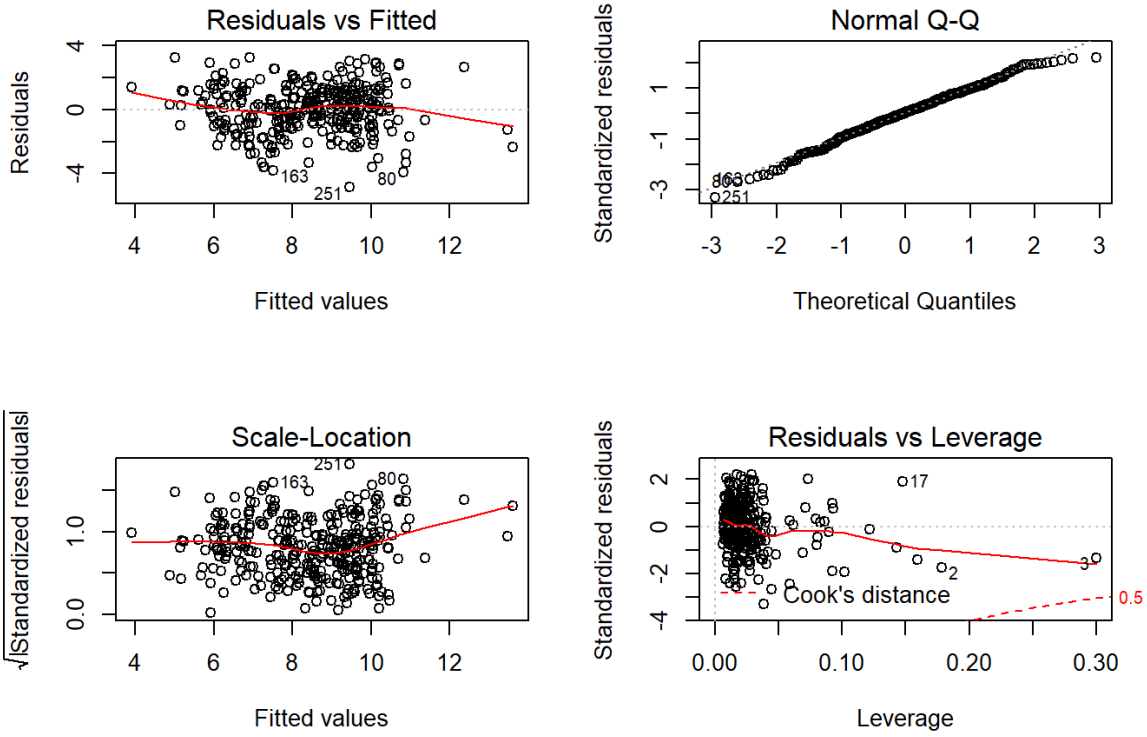
```
# remove 121 and 158 observation
reg_trans2 = lm(y_bc~x1+x2+x3+x4+x5+x6+x8, data = data_bc[-c(121,158),])
vif(reg_trans2)
```

```
##       x1       x2       x3       x4       x5       x6       x8
## 2.337850 1.101242 7.770519 3.375270 6.734428 4.628642 1.350872
```

```
summary(reg_trans2)  ###0.4969
```

```
##
## Call:
## lm(formula = y_bc ~ x1 + x2 + x3 + x4 + x5 + x6 + x8, data = data_bc[-c(121,
##     158), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8603 -0.9298  0.0868  1.0516  3.2509
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.080e+01  3.152e+00   3.427 0.000692 ***
## x1          -5.504e-04  2.206e-04  -2.495 0.013126 *
## x2           6.821e-05  2.692e-05   2.534 0.011779 *
## x3           1.178e-01  4.753e-02   2.477 0.013774 *
## x4          -1.037e-01  2.078e-02  -4.988 1.02e-06 ***
## x5           2.692e-01  3.591e-02   7.496 6.98e-13 ***
## x6          -1.262e-03  2.547e-04  -4.957 1.18e-06 ***
## x8          -3.610e-01  2.253e-01  -1.602 0.110166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.499 on 309 degrees of freedom
## Multiple R-squared:  0.5081, Adjusted R-squared:  0.4969
## F-statistic: 45.59 on 7 and 309 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(reg_trans2)
```




```
# stat test
ncvTest(reg_trans2)  ### 0.40802
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.6845703, Df = 1, p = 0.40802
```

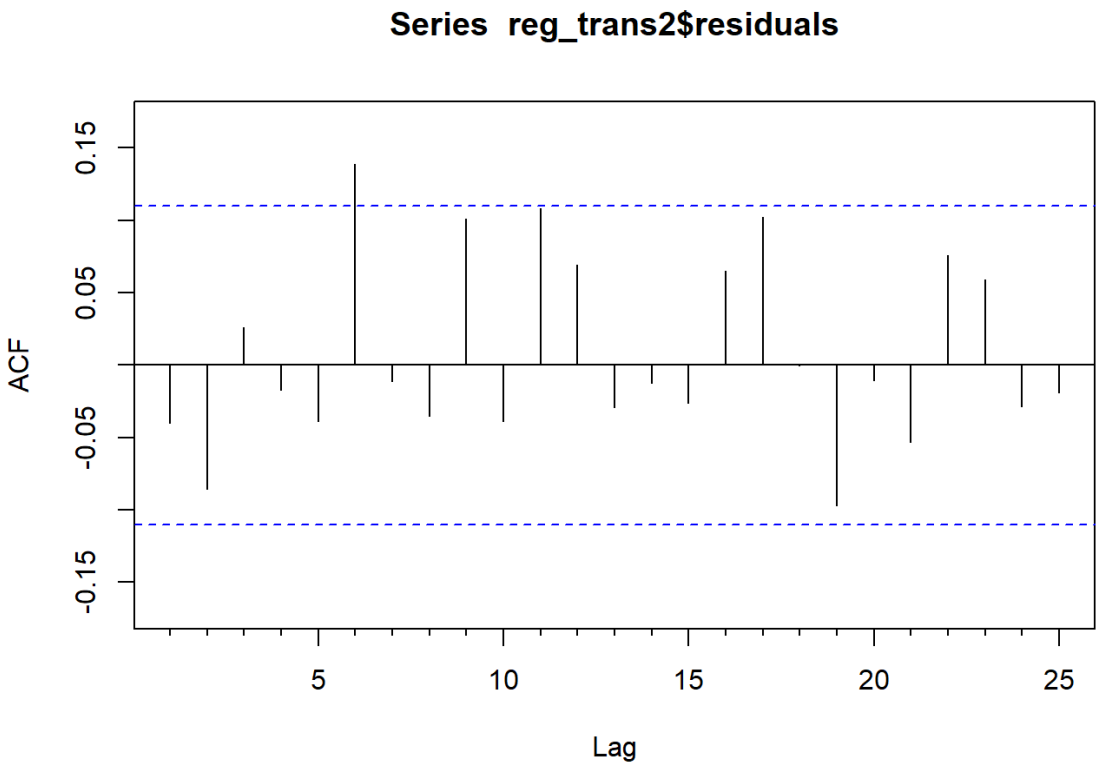
```
durbinWatsonTest(reg_trans2)   ###0.48
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.04023716      2.079303   0.482
##  Alternative hypothesis: rho != 0
```

```
shapiro.test(reg_trans2$residuals)   ###0.1169
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reg_trans2$residuals
## W = 0.99259, p-value = 0.1169
```

```
par(mfrow=c(1,1))
Acf(reg_trans2$residuals)
```



According to these result, it shows that there is multicollinearity in transformed dataset, because vif value of x3 and x5 are greater than 5. The summary information shows that 50.81% of variances in the dataset could be explained by estimated model which better than before. This regression model, all predictors except x8 are significant at the 5% level of significance.

The residuals test shows that the residuals around zero mean level until point 11 for fitted value, there is no auto-correlation at first lag while appear at sixth lag. However, the constant variance and normally distributed are not violated for these residuals. The residuals and leverage plot show that the observation 17 outside of red dash line, this might be the outlier. Through the test plot, it seems that the variance is not quite constant, and some of values not distributed normally, these result could be seen from statistic test as well, the p-value is greater than 0.05 slightly, but still could not reject null hypothesis.

4.2.2.3 Model without predictor x8 and observation 17 as well

Based on models fitted before, it shows that x8 is not significant at the 5% level of significance on transformed data. Trying remove not significant predictor and outlier (observation 17,121 and 158).

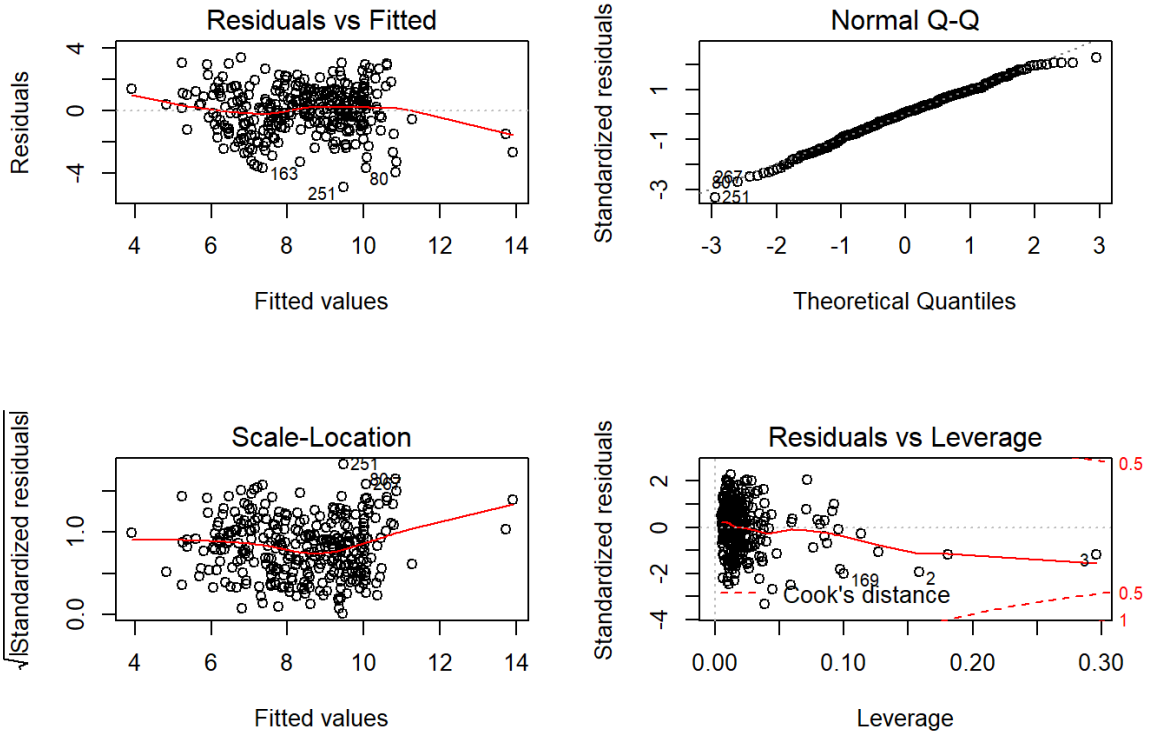
```
# remove x8, observation 17
reg_trans_mult = lm(y_bc~x1+x2+x3+x4+x5+x6, data = data_bc[-c(17,121,158),])
vif(reg_trans_mult)
```

```
##       x1       x2       x3       x4       x5       x6
## 2.336128 1.093651 7.611036 2.883347 6.684229 4.487622
```

```
summary(reg_trans_mult)  ###0.4851
```

```
##
## Call:
## lm(formula = y_bc ~ x1 + x2 + x3 + x4 + x5 + x6, data = data_bc[-c(17,
##     121, 158), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8769 -0.9452  0.0931  1.0511  3.3765
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.228e+01  3.009e+00   4.080 5.74e-05 ***
## x1          -4.878e-04  2.204e-04  -2.214   0.0276 *
## x2           4.205e-05  2.859e-05   1.471   0.1424
## x3           1.049e-01  4.694e-02   2.236   0.0261 *
## x4          -1.158e-01  1.916e-02  -6.045 4.30e-09 ***
## x5           2.705e-01  3.579e-02   7.559 4.65e-13 ***
## x6          -1.337e-03  2.502e-04  -5.345 1.76e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.496 on 309 degrees of freedom
## Multiple R-squared:  0.4949, Adjusted R-squared:  0.4851
## F-statistic: 50.45 on 6 and 309 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(reg_trans_mult)
```




```
# stat test
ncvTest(reg_trans_mult)  ### 0.48722
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.4826583, Df = 1, p = 0.48722
```

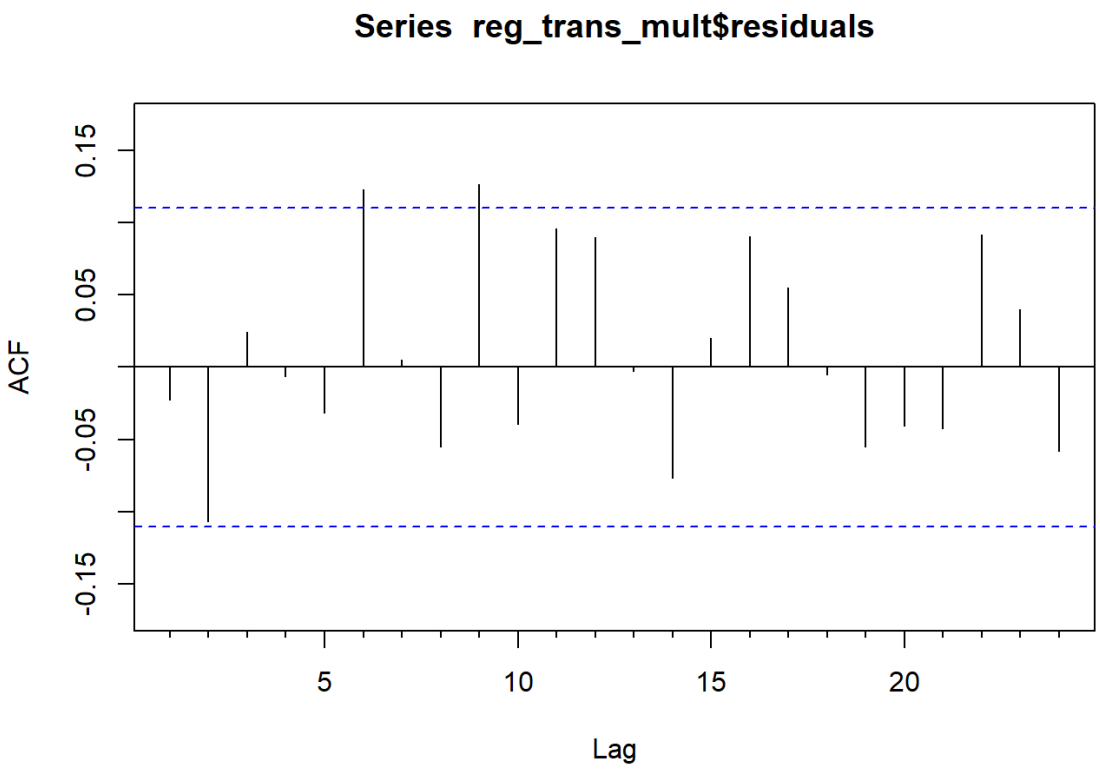
```
durbinWatsonTest(reg_trans_mult)   ###0.7
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.02288355      2.044566   0.686
##  Alternative hypothesis: rho != 0
```

```
shapiro.test(reg_trans_mult$residuals)   ###0.1355
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reg_trans_mult$residuals
## W = 0.99284, p-value = 0.1355
```

```
par(mfrow=c(1,1))
Acf(reg_trans_mult$residuals)
```



According to these result, it shows that there is multicollinearity in transformed dataset because vif values of x3 and x5 are greater than 5. The summary information shows that 49.49% of variances in the dataset could be explained by estimated model. This regression model, x1,x3,x4,x5,x6 are significant at the 5% level of significance while x2 not significant for estimated model.
The residuals test shows that the residuals around zero mean level until point 11 for fitted value, there is no auto-correlation at first lag while appear at sixth and ninth lag. However, the constant variance and normally distributed are not violated for these residuals.

4.2.2.4 Model with predictors x1, x3, x4, x5 and x6

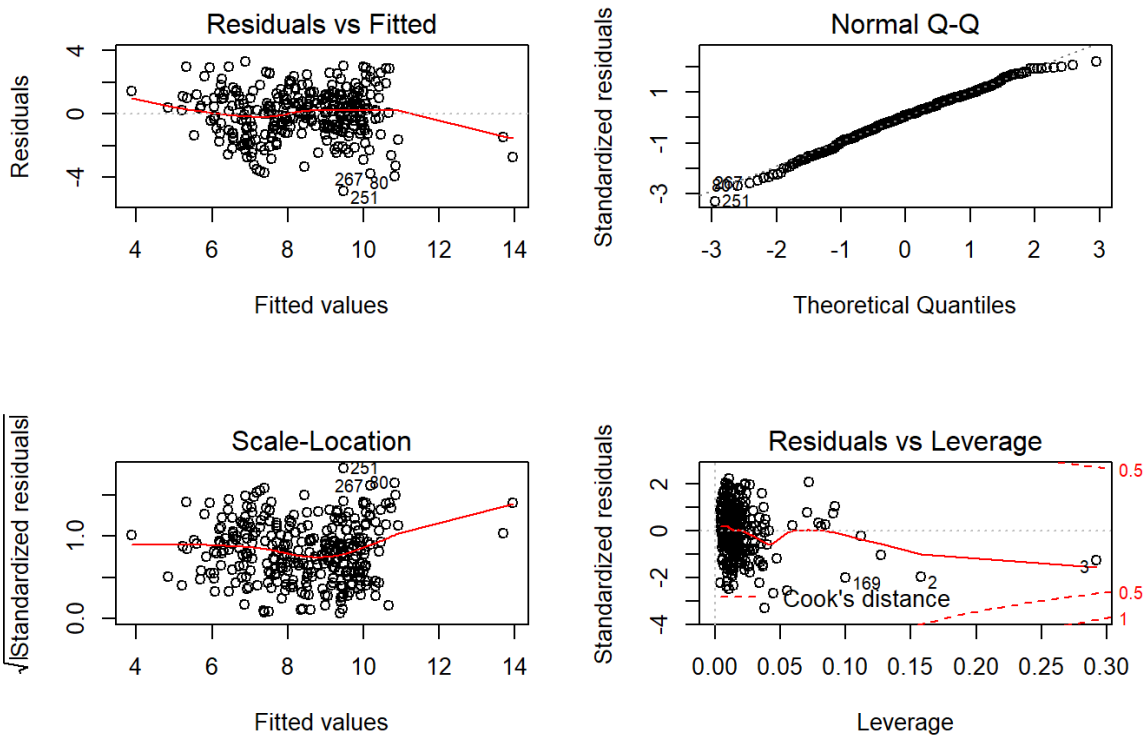
```
## remove x2 as well
reg_trans_mult2 = lm(y_bc~x1+x3+x4+x5+x6, data = data_bc[-c(17,121,158),])
vif(reg_trans_mult2)
```

```
##       x1       x3       x4       x5       x6
## 2.271827 7.531205 2.883018 6.557327 4.487574
```

```
summary(reg_trans_mult2)   ###0.4831
```

```
##
## Call:
## lm(formula = y_bc ~ x1 + x3 + x4 + x5 + x6, data = data_bc[-c(17,
##     121, 158), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8730 -0.9357  0.1131  1.0165  3.2916
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.9387022  3.0058695   3.972 8.87e-05 ***
## x1          -0.0004341  0.0002177  -1.994   0.0471 *
## x3           0.1120060  0.0467793   2.394   0.0172 *
## x4          -0.1155401  0.0191981  -6.018 4.97e-09 ***
## x5           0.2777884  0.0355133   7.822 8.22e-14 ***
## x6          -0.0013385  0.0002507  -5.340 1.80e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.498 on 310 degrees of freedom
## Multiple R-squared:  0.4913, Adjusted R-squared:  0.4831
## F-statistic: 59.89 on 5 and 310 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(reg_trans_mult2)
```




```
# stat test
ncvTest(reg_trans_mult2)  ### 0.41596
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.6616936, Df = 1, p = 0.41596
```

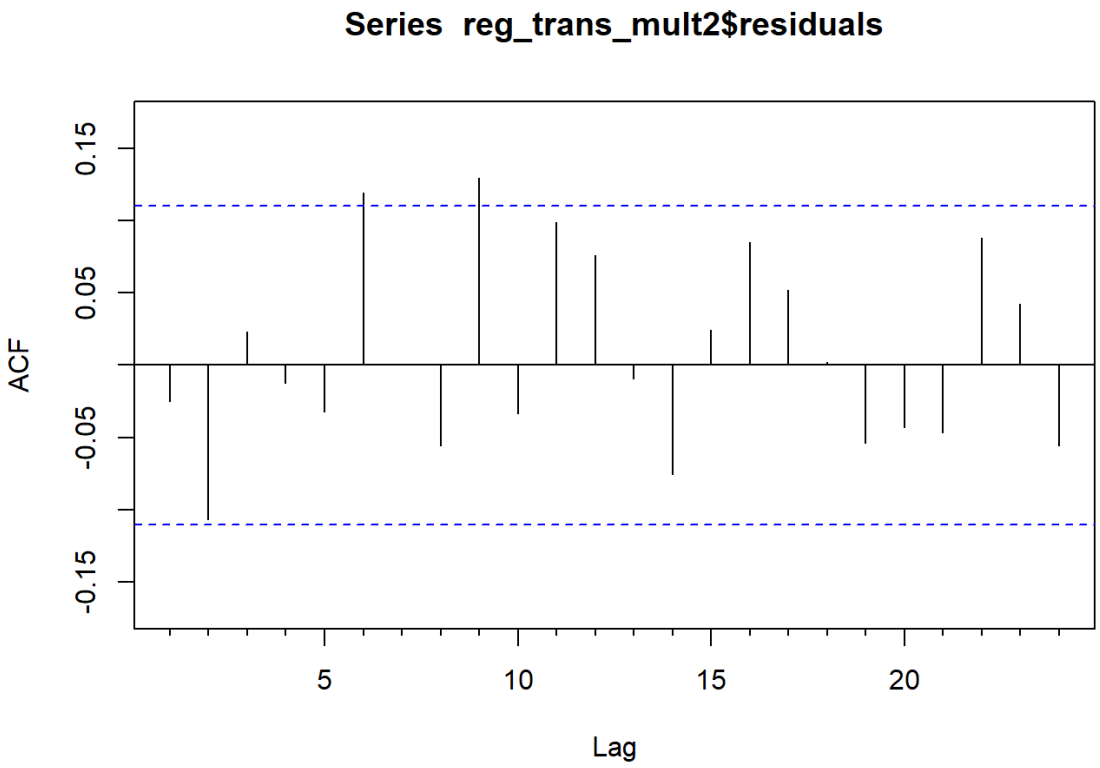
```
durbinWatsonTest(reg_trans_mult2)   ###0.682
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.02510661      2.049021    0.68
##  Alternative hypothesis: rho != 0
```

```
shapiro.test(reg_trans_mult2$residuals)    ###0.07416
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reg_trans_mult2$residuals
## W = 0.99172, p-value = 0.07416
```

```
par(mfrow=c(1,1))
Acf(reg_trans_mult2$residuals)
```



According to these result, it shows that there is slight multicollinearity in transformed dataset, because vif values of x3 and x5 are greater than 5.
The summary information shows that 49.13% of variances could be explained by estimated model which worse than before one. This regression model, all predictors are significant at the 5% level of significance.

The residuals test shows that the residuals around zero mean level until point 11 for fitted value, there is no auto-correlation at first lag while appear at sixth and ninth lag. However, the constant variance is not violated while normally distributed is violated for these residuals.

4.2.2.5 Model predictors x1, x3, x4, x5 and x6, without intercept

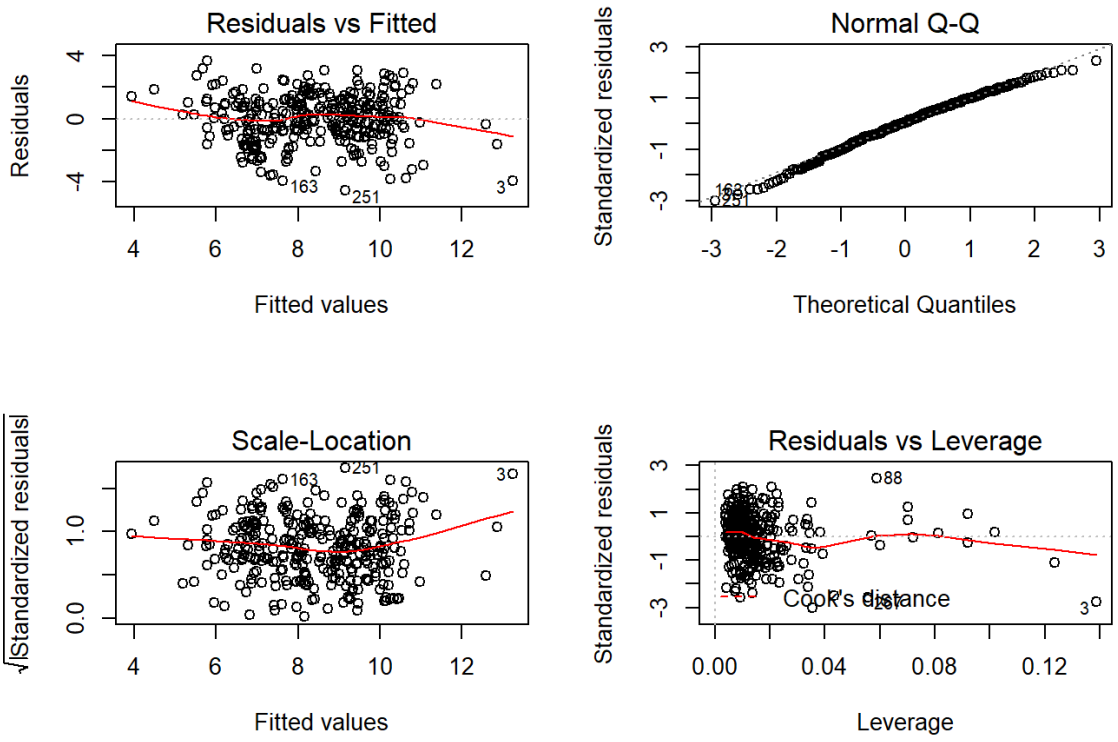
```
## remove intercept
reg_trans_mult3 = lm(y_bc~x1+x3+x4+x5+x6-1, data = data_bc[-c(17,121,158),])
vif(reg_trans_mult3)
```

```
##         x1         x3         x4         x5         x6
##  10.232946  40.211076 271.656876 120.041070   1.311618
```

```
summary(reg_trans_mult3)   ###0.9686
```

```
##
## Call:
## lm(formula = y_bc ~ x1 + x3 + x4 + x5 + x6 - 1, data = data_bc[-c(17,
##     121, 158), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5453 -0.9249  0.0700  1.0562  3.6676
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## x1 -0.0004339  0.0002229  -1.947 0.052453 .
## x3  0.2482974  0.0325394   7.631 2.88e-13 ***
## x4 -0.0564651  0.0124234  -4.545 7.87e-06 ***
## x5  0.3617504  0.0292062  12.386  < 2e-16 ***
## x6 -0.0004649  0.0001231  -3.778 0.000189 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.534 on 311 degrees of freedom
## Multiple R-squared:  0.9691, Adjusted R-squared:  0.9686
## F-statistic:  1952 on 5 and 311 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(reg_trans_mult3)
```




```
# stat test
ncvTest(reg_trans_mult3)  ### 0.50394
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.4466382, Df = 1, p = 0.50394
```

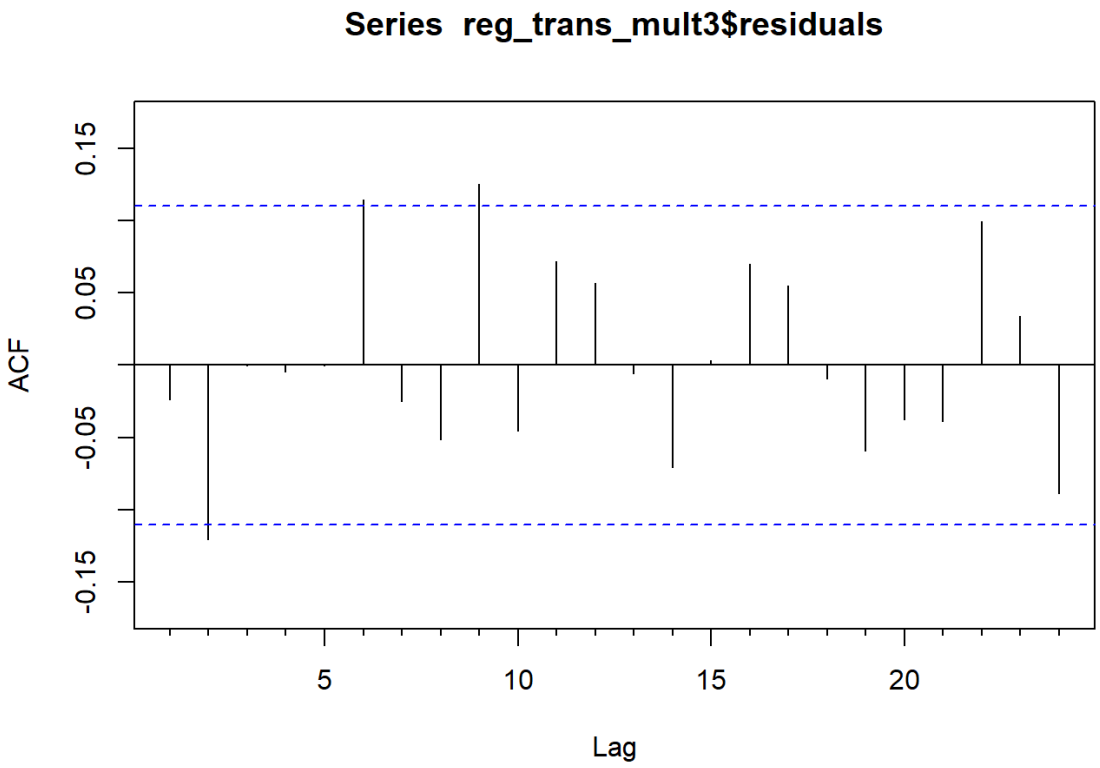
```
durbinWatsonTest(reg_trans_mult3)   ###0.698
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.02352492       2.04535   0.644
##  Alternative hypothesis: rho != 0
```

```
shapiro.test(reg_trans_mult3$residuals)    ###0.04147
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reg_trans_mult3$residuals
## W = 0.99064, p-value = 0.04147
```

```
par(mfrow=c(1,1))
Acf(reg_trans_mult3$residuals)
```



According to these result, it shows that there is serious multicollinearity in transformed dataset because most of vif values are greater than 10 much.
The summary information shows that 96.91% of variances could be explained by estimated model which much better than before models. This regression model, all predictors are significant at the 5% level of significance.

The residuals test shows that the residuals around zero mean level until point 11 of fitted value, there is no auto-correlation at first lag while appear at sixth and ninth lag. The constant variance is not violated while the normally distributed is violated for residuals.

Comparing this estimated model and before one, the difference is that whether include intercept or not, this one without intercept increase value of adjusted R-squared sharply, but the residuals violate normally distributed assumptions, and the multicollinearty is the big issue for fitted model. In fact, in the model with intercept, trying remove x3 to solve the slight multicollinearity issue, the result shows that all of assumptions are violated, so, just keep both x3 and x5 in fitted model, although the slight multicollinearity issue in it.

4.2.2.6 Weighted least square model without predictors x2, x7, x8, x9

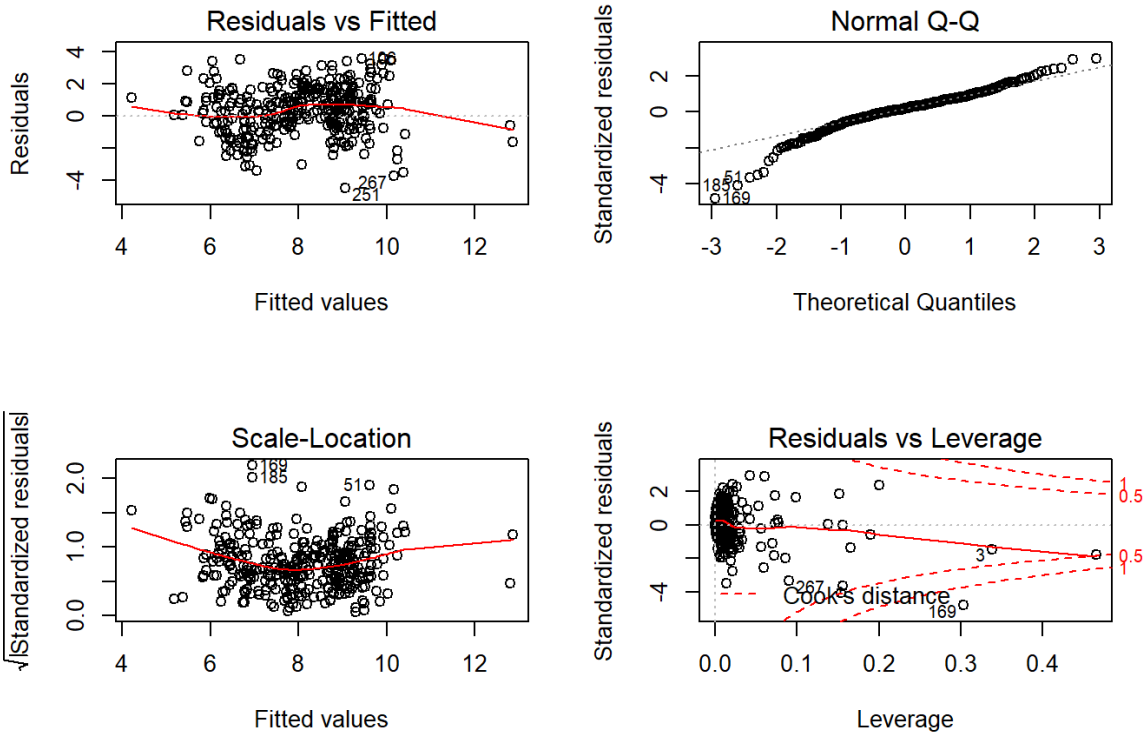
```
# weighted least square model
reg_trans_mult_w = lm(y_bc~x1+x3+x4+x5+x6, weights = 1/x2,data = data_bc[-c(17,121,158),])
vif(reg_trans_mult_w)
```

```
##       x1       x3       x4       x5       x6
## 2.721955 5.790498 5.547765 3.716493 8.008138
```

```
summary(reg_trans_mult_w)  ###0.5261
```

```
##
## Call:
## lm(formula = y_bc ~ x1 + x3 + x4 + x5 + x6, data = data_bc[-c(17,
##     121, 158), ], weights = 1/x2)
##
## Weighted Residuals:
##       Min        1Q    Median        3Q       Max
## -0.208361 -0.017588  0.009618  0.035993  0.149456
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.174e+00  2.083e+00   4.403 1.47e-05 ***
## x1           6.482e-06  2.315e-04   0.028 0.977680
## x3           1.068e-01  2.983e-02   3.580 0.000399 ***
## x4          -9.872e-02  1.605e-02  -6.149 2.39e-09 ***
## x5           2.770e-01  2.659e-02  10.417  < 2e-16 ***
## x6          -9.893e-04  1.647e-04  -6.009 5.24e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.052 on 310 degrees of freedom
## Multiple R-squared:  0.5336, Adjusted R-squared:  0.5261
## F-statistic: 70.93 on 5 and 310 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(reg_trans_mult_w)
```




```
# stat test
ncvTest(reg_trans_mult_w)  ### 0.061022
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 3.509396, Df = 1, p = 0.061022
```

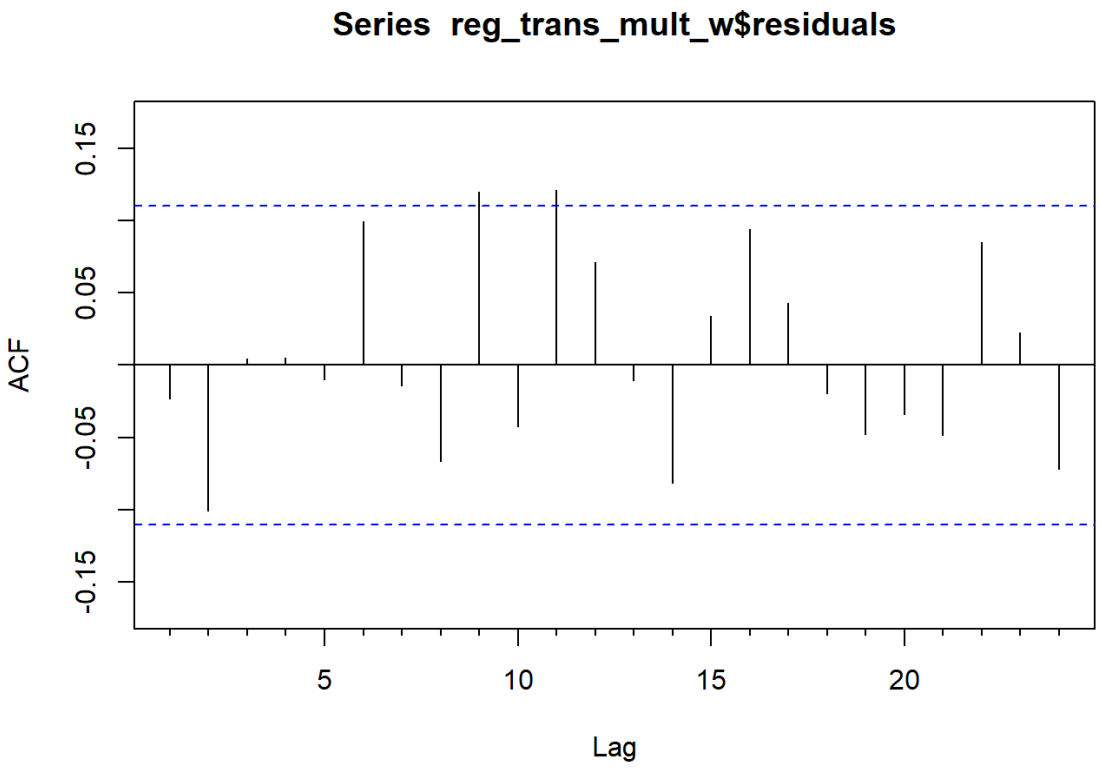
```
durbinWatsonTest(reg_trans_mult_w)  ###0.596
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.02775855      1.942986   0.618
##  Alternative hypothesis: rho != 0
```

```
shapiro.test(reg_trans_mult_w$residuals)    ###0.08732
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reg_trans_mult_w$residuals
## W = 0.99202, p-value = 0.08732
```

```
par(mfrow=c(1,1))
Acf(reg_trans_mult_w$residuals)
```



According to these result, it shows that there is multicollinearity in this fitted model, because vif of x3, x4 and x6 are greater than 5.
The summary information shows that 53.36% of variances could be explained by estimated model. This regression model, x3,x4,x5,x6 are significant at the 5% level of significance while x1 not significant for estimated model.

The residuals test shows that the residuals not around zero mean level, there is no auto-correlation at first lag while appear at ninth and eleventh lag. However, the constant variance and normally distributed are not violated.

4.2.2.7 Transformation with 0-1 scale

```
# 0-1 scale data on data without multicollinearity
minmaxnormalise = function(x){
  (x-min(x)) / (max(x)-min(x))
}
data_no_mul = data_no_city
data_minmax = as.data.frame(lapply(data_no_mul,minmaxnormalise))
head(data_minmax,3)
```

```
##            y         x1          x2        x3          x4        x5
## 1 0.03873239 0.25144509 0.009968131 0.3572866 0.446911403 0.4410181
## 2 0.44014085 0.01003303 0.023554184 0.4934132 0.003183904 0.7401255
## 3 0.25704225 0.03876961 0.012030876 0.8915470 0.517715791 0.6653942
##          x6           x7 x8        x9         x10
## 1 0.5820235 0.0003861153  0 0.4122514 0.03133666
## 2 0.2479522 0.2527510715  0 0.3823487 0.03133666
## 3 0.3730352 0.0000000000  0 0.4122514 0.03133666
```

```
reg_minmax_y = lm(y~., data = data_minmax)
summary(reg_minmax_y)  ###0.4381
```

```
##
## Call:
## lm(formula = y ~ ., data = data_minmax)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34260 -0.06732 -0.01202  0.05874  0.45007
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02429    0.12859   0.189  0.85027
## x1          -0.13254    0.04057  -3.267  0.00121 **
## x2           0.12204    0.11652   1.047  0.29573
## x3           0.32495    0.10702   3.036  0.00260 **
## x4          -0.22831    0.07257  -3.146  0.00182 **
## x5           0.56330    0.08408   6.700 9.93e-11 ***
## x6          -0.25504    0.08574  -2.974  0.00317 **
## x7          -0.02437    0.06136  -0.397  0.69148
## x8          -0.02122    0.01728  -1.228  0.22029
## x9          -0.05172    0.07338  -0.705  0.48145
## x10          0.05534    0.11383   0.486  0.62717
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1138 on 308 degrees of freedom
## Multiple R-squared:  0.4557, Adjusted R-squared:  0.4381
## F-statistic: 25.79 on 10 and 308 DF,  p-value: < 2.2e-16
```

```
# step
step(reg_minmax_y, direction = 'both')
```

```
## Start:  AIC=-1375.59
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10
##
##        Df Sum of Sq    RSS     AIC
## - x7    1   0.00204 3.9930 -1377.4
## - x10   1   0.00306 3.9940 -1377.3
## - x9    1   0.00644 3.9974 -1377.1
## - x2    1   0.01422 4.0052 -1376.5
## - x8    1   0.01955 4.0105 -1376.0
## <none>              3.9910 -1375.6
## - x6    1   0.11463 4.1056 -1368.6
## - x3    1   0.11945 4.1104 -1368.2
## - x4    1   0.12825 4.1192 -1367.5
## - x1    1   0.13829 4.1292 -1366.7
## - x5    1   0.58162 4.5726 -1334.2
##
## Step:  AIC=-1377.43
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x8 + x9 + x10
##
##        Df Sum of Sq    RSS     AIC
## - x10   1   0.00239 3.9954 -1379.2
## - x9    1   0.00675 3.9998 -1378.9
## - x2    1   0.01444 4.0074 -1378.3
## - x8    1   0.01885 4.0119 -1377.9
## <none>              3.9930 -1377.4
## + x7    1   0.00204 3.9910 -1375.6
## - x6    1   0.11273 4.1057 -1370.5
## - x3    1   0.12325 4.1163 -1369.7
## - x4    1   0.12630 4.1193 -1369.5
## - x1    1   0.13636 4.1294 -1368.7
## - x5    1   0.60763 4.6006 -1334.2
##
## Step:  AIC=-1379.24
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x8 + x9
##
##        Df Sum of Sq    RSS     AIC
## - x9    1   0.00661 4.0020 -1380.7
## - x8    1   0.01851 4.0139 -1379.8
## <none>              3.9954 -1379.2
## + x10   1   0.00239 3.9930 -1377.4
## + x7    1   0.00137 3.9940 -1377.3
## - x6    1   0.11085 4.1062 -1372.5
## - x3    1   0.12580 4.1212 -1371.3
## - x4    1   0.13001 4.1254 -1371.0
## - x1    1   0.13469 4.1301 -1370.7
## - x2    1   0.13922 4.1346 -1370.3
## - x5    1   0.60981 4.6052 -1335.9
##
## Step:  AIC=-1380.71
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x8
##
##        Df Sum of Sq    RSS     AIC
## - x8    1   0.02401 4.0260 -1380.8
## <none>              4.0020 -1380.7
## + x9    1   0.00661 3.9954 -1379.2
## + x10   1   0.00225 3.9998 -1378.9
## + x7    1   0.00164 4.0004 -1378.8
## - x6    1   0.10952 4.1115 -1374.1
## - x3    1   0.12291 4.1249 -1373.1
## - x4    1   0.13132 4.1333 -1372.4
## - x1    1   0.13479 4.1368 -1372.1
## - x2    1   0.14833 4.1503 -1371.1
## - x5    1   0.60871 4.6107 -1337.5
##
## Step:  AIC=-1380.8
## y ~ x1 + x2 + x3 + x4 + x5 + x6
##
##        Df Sum of Sq    RSS     AIC
```

```
## <none>              4.0260 -1380.8
## + x8    1   0.02401 4.0020 -1380.7
## + x9    1   0.01210 4.0139 -1379.8
## + x10   1   0.00182 4.0242 -1379.0
## + x7    1   0.00108 4.0249 -1378.9
## - x3    1   0.11102 4.1370 -1374.1
## - x6    1   0.12777 4.1538 -1372.8
## - x1    1   0.12935 4.1554 -1372.7
## - x2    1   0.13685 4.1629 -1372.1
## - x4    1   0.20048 4.2265 -1367.3
## - x5    1   0.61884 4.6449 -1337.2
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6, data = data_minmax)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4
##     0.02119     -0.12723      0.16428      0.30826     -0.26238
##          x5           x6
##     0.57318     -0.26165
```

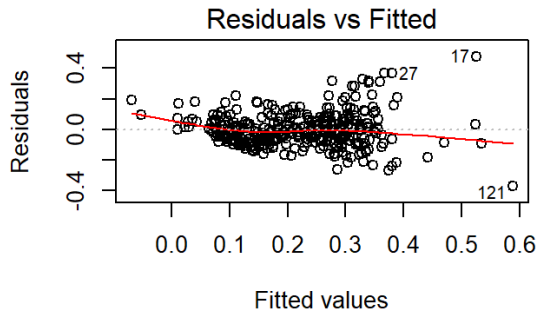
```
reg_minmax_y_step = lm(y~x1+x2+x3+x4+x5+x6, data = data_minmax)
summary(reg_minmax_y_step)  ###0.4404
```

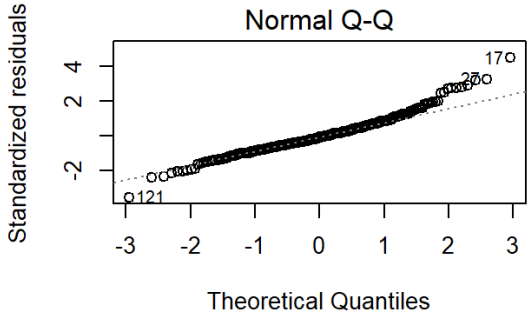
```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6, data = data_minmax)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37006 -0.07023 -0.01069  0.05610  0.47494
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02119    0.12067   0.176  0.86069
## x1          -0.12723    0.04019  -3.166  0.00170 **
## x2           0.16428    0.05044   3.257  0.00125 **
## x3           0.30826    0.10509   2.933  0.00360 **
## x4          -0.26238    0.06657  -3.942 9.99e-05 ***
## x5           0.57318    0.08277   6.925 2.49e-11 ***
## x6          -0.26165    0.08315  -3.147  0.00181 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1136 on 312 degrees of freedom
## Multiple R-squared:  0.4509, Adjusted R-squared:  0.4404
## F-statistic: 42.71 on 6 and 312 DF,  p-value: < 2.2e-16
```

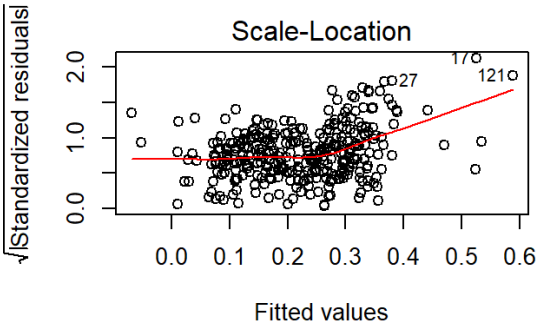
```
vif(reg_minmax_y_step)
```

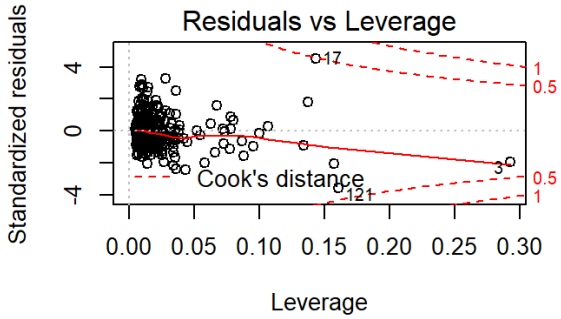
```
##       x1       x2       x3       x4       x5       x6
## 2.332504 1.085510 7.758788 2.651102 6.595683 4.661584
```

```
par(mfrow=c(2,2))
plot(reg_minmax_y_step)
```










```
# stat test
ncvTest(reg_minmax_y_step)  ### p < 2.22e-16
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 78.1626, Df = 1, p = < 2.22e-16
```

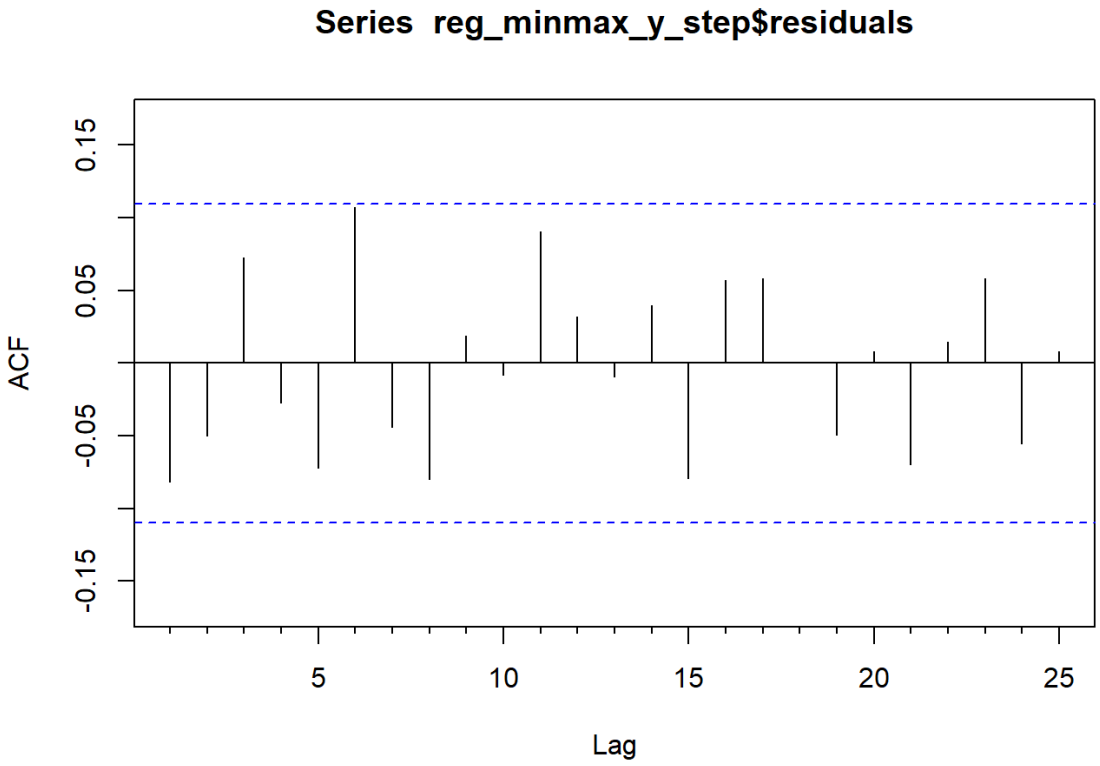
```
durbinWatsonTest(reg_minmax_y_step)   ###0.126
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      -0.08202058       2.163523   0.136
##  Alternative hypothesis: rho != 0
```

```
shapiro.test(reg_minmax_y_step$residuals)   ###1.794e-06
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reg_minmax_y_step$residuals
## W = 0.96818, p-value = 1.794e-06
```

```
par(mfrow=c(1,1))
Acf(reg_minmax_y_step$residuals)
```



Firstly, tryig fitted model with all predictors on 0-1 scale dataset, then, using step function to get best model to test assumption on best model based on this transformed dataset. According to these results, it shows that there is a slight multicollinearity in this fitted model.
The summary information shows that 45.09% of variances could be explained by estimated model. This regression model, all predictors are significant at the 5% level of significance.

The residuals test shows that the residuals around zero mean level, there is no auto-correlation. However, the constant variance and normally distributed are violated for residuals.

4.2.2.8 Fitting GLM model

It seems that transformed data does not work well on fitted model, there are still some assumptions are violated by fitted models. Tring glm() as well to see the performance.

```
lm1 = glm(y~., data = data_no_city)
summary.glm(lm1)
```

```
##
## Call:
## glm(formula = y ~ ., data = data_no_city)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -97.297  -19.120   -3.415   16.681  127.820
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.6407401 67.4450741   0.588  0.55713
## x1          -0.0155411  0.0047571  -3.267  0.00121 **
## x2           0.0013896  0.0013267   1.047  0.29573
## x3           3.0815651  1.0149309   3.036  0.00260 **
## x4          -1.3100712  0.4164184  -3.146  0.00182 **
## x5           5.1629261  0.7706162   6.700 9.93e-11 ***
## x6          -0.0160350  0.0053911  -2.974  0.00317 **
## x7          -0.0002673  0.0006728  -0.397  0.69148
## x8          -6.0270241  4.9070156  -1.228  0.22029
## x9          -0.2132142  0.3025039  -0.705  0.48145
## x10          0.0229407  0.0471839   0.486  0.62717
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1045.112)
##
##     Null deviance: 591418  on 318  degrees of freedom
## Residual deviance: 321894  on 308  degrees of freedom
## AIC: 3135.7
##
## Number of Fisher Scoring iterations: 2
```

```
deviance(lm1)
```

```
## [1] 321894.5
```

```
pchisq(lm1$deviance, df = lm1$df.residual, lower.tail = FALSE)
```

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

According to these result, it shows that the p-value of fitted model is 0, we should reject null hypothesis, which means that this fitted model is not good for this dataset.

5 Prediction

Based on models we fitted before, using reg_trans2 to predict AQI for test data.

5.0.1 nthroot function - convert transformed data into original data

```
nth_root <- function(A, n, tol=sqrt(.Machine$double.eps))
{
  ifelse(A < 1, x0 <- A * n, x0 <- A / n)
  repeat
  {
    x1 <- ((n-1)*x0 + A / x0^(n-1))/n
    if(abs(x1 - x0) > tol) x0 <- x1 else break
  }
  x1
}
magic_for(print,silent = TRUE)
```

5.0.2
x1=807.8,x2=1270.4,x3=16.18219178,x4=104.65019,x5=30.122671,x6=367,x7=178
y=86

```
##y=86
pred1 = predict(reg_trans2,data.frame(x1=807.8,x2=1270.4,x3=16.18219178,
                                      x4=104.65019,x5=30.122671,x6=367,
                                      x7=1787,x8=0,x9=38.67,x10=18.28),
        interval="prediction", level = 0.95)

for (i in pred1){
  pred_real1=(nth_root(i*lambda+1, lambda))
  print(pred_real1)
}
pred_1=as.data.frame(t(magic_result_as_vector()))
colnames(pred_1) = c('pred_real_fit','pred_real_lwr','pred_real_upr')
pred_1
```

```
##   pred_real_fit pred_real_lwr pred_real_upr
## 1       81.4912      32.83082       166.344
```

5.0.3
x1=288,x2=4130.2,x3=14.57671233,x4=118.0560532,x5=36.7935791,x6=38,x7=782
y=116

```
##y=116
pred2 = predict(reg_trans2,data.frame(x1=288,x2=4130.2,x3=14.57671233,
                                      x4=118.0560532,x5=36.7935791,x6=38,
                                      x7=782,x8=0,x9=36,x10=31),
        interval="prediction", level = 0.95)
for (i in pred2){
  pred_real2=(nth_root(i*lambda+1, lambda))
  print(pred_real2)
}
pred_2=as.data.frame(t(magic_result_as_vector()))
colnames(pred_2) = c('pred_real_fit','pred_real_lwr','pred_real_upr')
pred_2
```

```
##   pred_real_fit pred_real_lwr pred_real_upr
## 1      108.3545      47.54845      209.6283
```

5.0.4
x1=994.8,x2=1143.11,x3=19.43287671,x4=104.7763519,x5=29.36772156,x6=311,x7
y=118

```
##y=118
pred3 = predict(reg_trans2,data.frame(x1=994.8,x2=1143.11,x3=19.43287671,
                                      x4=104.7763519,x5=29.36772156,x6=311,
                                      x7=1557,x8=0,x9=40.2,x10=35.47),
        interval="prediction", level = 0.95)
for (i in pred3){
  pred_real3=(nth_root(i*lambda+1, lambda))
  print(pred_real3)
}
pred_3=as.data.frame(t(magic_result_as_vector()))
colnames(pred_3) = c('pred_real_fit','pred_real_lwr','pred_real_upr')
pred_3
```

```
##   pred_real_fit pred_real_lwr pred_real_upr
## 1      84.25573      34.33221      170.7878
```

5.0.5
x1=1000,x2=2168.34,x3=16.99178082,x4=106.9293976,x5=27.69538689,x6=865,x7
y=60

```
##y=60
pred4 = predict(reg_trans2,data.frame(x1=1000,x2=2168.34,x3=16.99178082,
                                      x4=106.9293976,x5=27.69538689,x6=865,
                                      x7=3581,x8=0,x9=44.06,x10=34.51),
        interval="prediction", level = 0.95)
for (i in pred4){
  pred_real4=(nth_root(i*lambda+1, lambda))
  print(pred_real4)
}
pred_4=as.data.frame(t(magic_result_as_vector()))
colnames(pred_4) = c('pred_real_fit','pred_real_lwr','pred_real_upr')
pred_4
```

```
##   pred_real_fit pred_real_lwr pred_real_upr
## 1      53.70358      18.89365      118.8285
```

6 Comparison

Comparing all of fitted models, it seems that model reg_trans2 with transformed dataset (BoxCox) without outliers observation 17,121 and 158, include predictors x1,x2,x3,x4,x5,x6 and x8 show the best performance. Although just 50.81% variances in the data could be explained by estimated model. The residuals are around zero mean level until point 11, constant error variance and normally distributed are not violated, just slight auto-correlation at 6th lag.

```
pred_trans=rbind(pred1,pred2,pred3,pred4)
pred_real=rbind(pred_1,pred_2,pred_3,pred_4)
pred_test=cbind(pred_trans,pred_real)
pred_test$City=c('Ziyang City','Zibo City','Zigong City','Zunyi City')
pred_test$AQI=c(86,116,118,60)
pred_test$accuracy=pred_test$pred_real_fit/pred_test$AQI
pred_test[c(7,8,1:6,9)] %>%
  knitr::kable(caption = 'Prediction AQI for Test Dataset',row.names = FALSE)
```

Prediction AQI for Test Dataset

City	AQI	fit	lwr	upr	pred_real_fit	pred_real_lwr	pred_real_upr	accuracy
Ziyang City	86	9.146591	6.167535	12.12565	81.49120	32.83082	166.3440	0.9475720
Zibo City	116	10.260217	7.284103	13.23633	108.35455	47.54845	209.6283	0.9340909
Zigong City	118	9.272123	6.295846	12.24840	84.25573	34.33221	170.7878	0.7140316
Zunyi City	60	7.679009	4.716252	10.64177	53.70358	18.89365	118.8285	0.8950597

Comparing test dataset on, the performance looks good, the average accuracy is 87.27%, although for some observation like Zigong City, the fitted model does not work well, most of them could got acceptable prediction result. In fact, in prediction section, the real AQI value inside of 95% confidence interval for each observation.

7 Summary

According to this project, different performance on different fitted models with different predictors or different dataset could be seen clearly. Although tried many fitted models, there is not one model is really good enough, because each model has its own issues, most of them have not constant variance, and not normally distributed for residuals. Generally, these issues could be done by transformation, it seems that not work well for this dataset. Weighted least squares is another method to solve these problem, but how to deal with weight is a difficult topic, this project just used simple weight to see the performance and it does not work well. One more thing is that observation 17, 121 and 158 as the outliers are removed when fitting models, whether these observation should be consider as outliers could be discussed further in the real life, cause these data might useful if there is not mistake with data itself. In fact, deal with outliers is an essential process when work with real data, and the issue like non-constant variance will also appear usually, trying to find good method or reasonable weight should be done further. One of the good way is that communicate with other senior analysts or related field experts.

References

Air Quality Index (AQI) Basics. 2016. airnow.gov. https://airnow.gov/index.cfm?action=aqibasics.aqi (https://airnow.gov/index.cfm?action=aqibasics.aqi).

Liu, Jianzheng, Weifeng Li, Jiansheng Wu, and Yonghong Liu. 2018. *Visualizing the Intercity Correlation of Pm2.5 Time Series in the Beijing-Tianjin-Hebei Region Using Ground-Based Air Quality Monitoring Data*. PLoS ONE. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0192614 (https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0192614).