STAT1006

Regression and Nonparametric Inference

Semester 2, 2024

Final Project Report

LINEAR REGRESSION ANALYSIS FOR SULFUR DIOXIDE CONTENT IN THE ATMOSPHERE

Bruce Omondi, 20074027

Bachelor of Science (Data Science)

Faculty of Science and Engineering |School of Electrical Engineering, Computing and Mathematical Sciences
CRICOS Provider Code 00301J

Declaration

The work presented in this report is my own work and all references are duly acknowledged.

This work has not been submitted, in whole or in part, in respect of any academic award at Curtin University or elsewhere.

Bruce Omondi

04/11/2024

Contents

1 Introduction

This study employs simple machine learning techniques, i.e., linear regression analysis, to investigate a dataset containing various factors, with the aim of coming up with the best predictive model to predict the amount of SO_2 content in the air from these factors. SO_2 levels are used as an indicator of the amount of air pollution in many urban areas, and high amounts of SO_2 in the air generally equate to high levels of air pollution, which can be harmful to people's health and the environment at large with possible respiratory problems and the formation of acid rain. This study would be helpful in understanding how different ecological and human factors affect the amount of air pollution and can help environmental scientists, among others, to plan ahead and put measures in place to control the amount of SO_2 emissions in urban areas.

The study aims to find out which of these factors contribute the most to the increase in SO_2 emissions, how well can we predict the amount of SO_2 content in the air with these explanatory factors, and which linear regression model would be best suited for this task. The report will cover the exploratory analysis of the dataset, simple linear regression with only one explanatory variable, and multiple linear regression with multiple explanatory variables and the SO_2 content as the target variable. Lastly, there will be a comparison of the predictive abilities of the different models to decide which one is the best predictive model.

The dataset is relatively small compared to the ideal size for regression and other machine learning applications, as it only contains forty one observations factored into seven variables – SO_2 , manu, temp, popul, wind, precip, predays, and Region:

1. **SO_2 :** mean amount of sulphur dioxide in the air (mg/cm^3). This is the variable we are trying to predict, i.e., the response variable.
2. **Temp:** mean annual temperature (in Fahrenheit)
3. **Manu:** number of manufacturing factories with 20 or more workers
4. **Popul:** the population size in thousands as of 1970 census

5. **Wind:** mean annual wind speed (in miles per hour)
6. **Precip:** mean annual amount of precipitation (in inches)
7. **Predays:** the average number of days with precipitation annually
8. **Region:** the region in question (in USA)

2 Exploratory Data Analysis (EDA)

To begin with, I checked the dataset for any issues that might need to be cleaned. All of the data entries were correct with no inconsistent entries or missing values in any of the columns, the variables had the right data types, and there were no duplicate records. I then conducted some exploratory data analysis to get a better understanding of the dataset and the variables. Firstly, I opened the dataset and a quick glimpse into the first six observations of the dataset also revealed the data types for each variable. All of the variables were numerical except Region, which was a categorical variable containing the region of each observation. I then checked the five-number summaries of the variables to gain some insight into the spread (quartiles), location (median), and range (minimum and maximum values) of each of the variables. Variable 'manu' had the largest range, and 'popul' had the largest mean value. I could also see how the variables would spread if they were plotted on a boxplot from the minimum value, the first quartile, the median (second quartile), third quartile, and the maximum value. Seeing as Region was the only categorical variable in the dataset, I wanted to see how the SO_2 levels vary by region, so I plotted the histograms for SO_2 by Region. The amount of SO_2 content in the air was highest in the East, followed by North, South, then the West, in descending order as shown in Fig. 1 below.

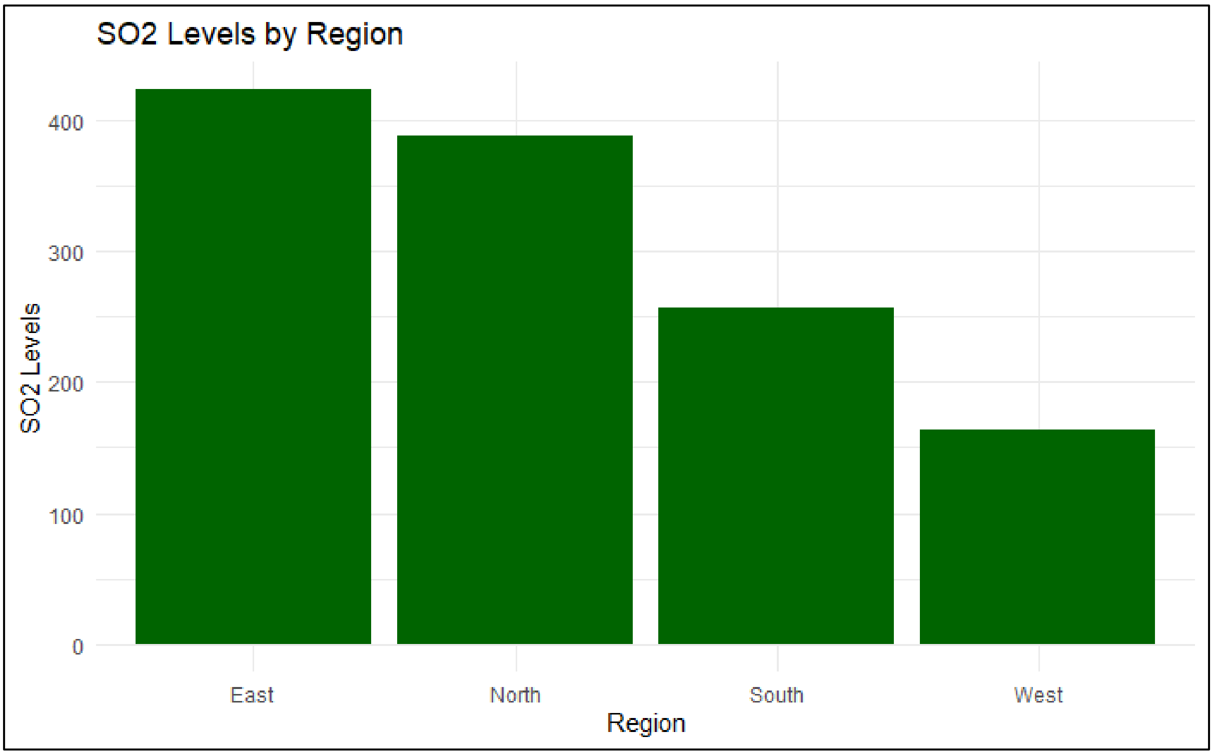


Fig. 1: Histogram of SO_2 content by Region

To get a better look at how the values were spread and possible outliers for the SO_2 values in each region, I plotted boxplots of the two variables. I could see that the SO_2 values in the East were right-skewed with no outliers, left-skewed in the North with two possible outliers, slightly right-skewed in the South with three possible outliers, and left-skewed in the West with one possible outlier, as shown in Fig.2 below.

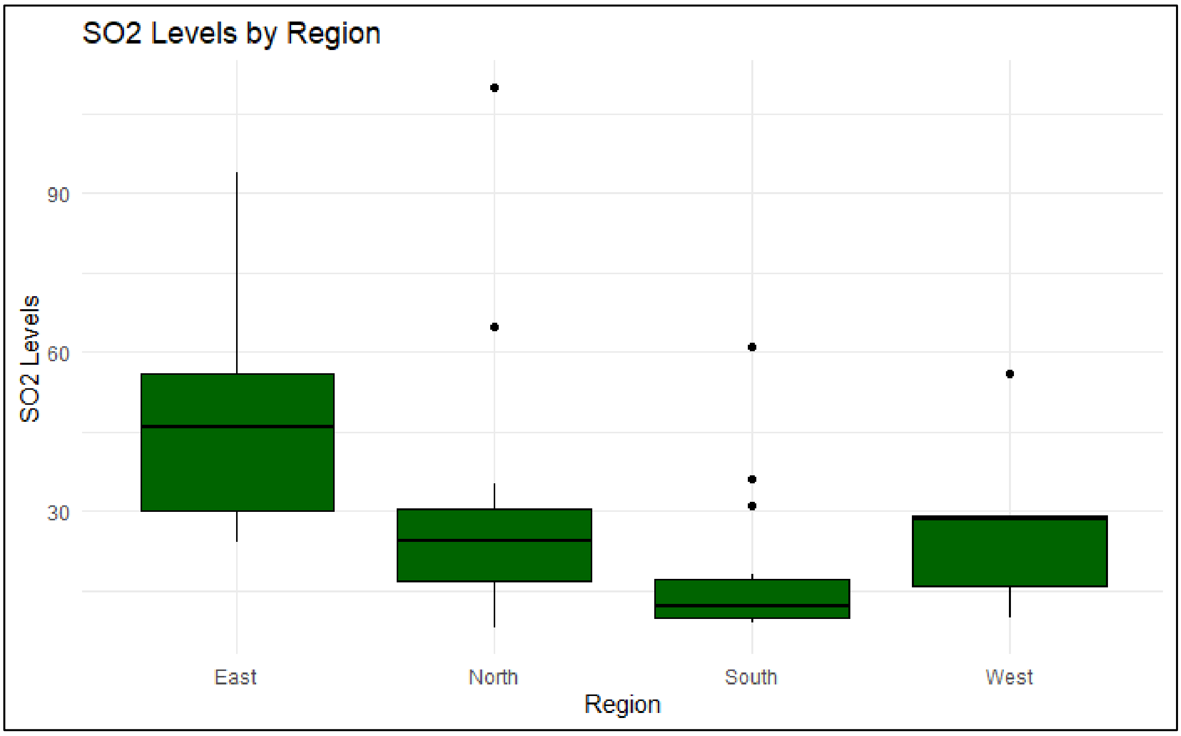


Fig. 2: Boxplot of SO_2 content by Region

I then executed a function to check for the number of possible outlier observations in each variable in the original dataset and found that SO_2 had 3, temp had 1, manu had 4, popul had 3, wind had no outliers, precip had 2, and predays had 3. Since the dataset was very small, I was able to manually check each of the outliers in each variable and confirmed that they were genuine datapoints, they just happened to be larger compared to the other observations in their respective variables. I therefore opted not to drop them. However, later on in the analysis, I did try to perform some transformations to see if they could help alleviate the influence of these potential outliers on the final model.

Lastly, to check for the association and correlation between the variables, I plotted the correlation plot. This provided great insight into which variable had the strongest correlation with the target variable (SO_2) for the subsequent simple linear regression model, the level of multicollinearity in the dataset, i.e., which variables had very strong correlations between themselves, and the nature of the relationship between the different variables. I also got the Pearson's correlation coefficient values for the correlation between each of the variables, and with the SO_2 target variable, which was crucial in deciding which variable would be best appropriate for modelling the simple linear regression model. The correlation plot is shown in Fig. 3 below.

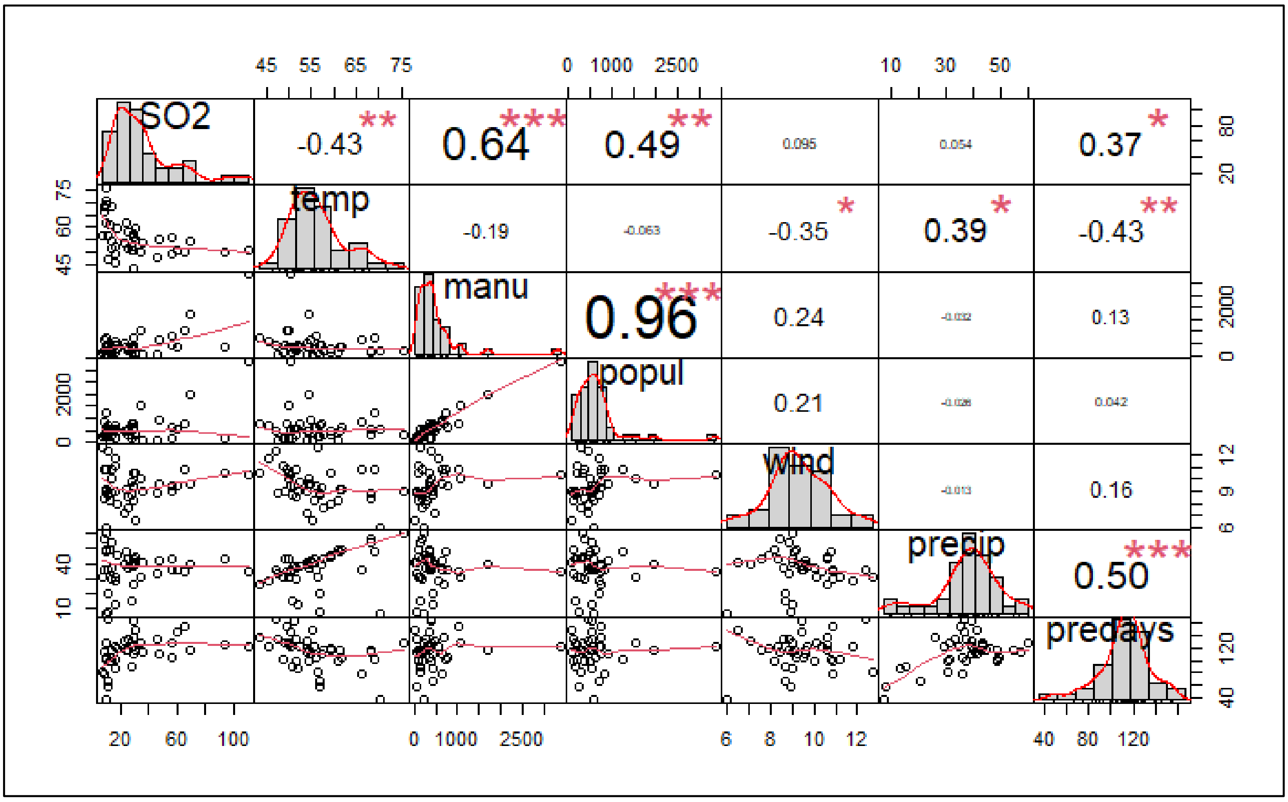


Fig. 3: Correlation plot

From the plot above, I could see that many of the variables did not fulfill the assumption of linearity with the response variable, SO_2. I explored different transformations to try and improve the linearity of the variable relationships, stabilize the variances, and possibly improve the normality of the data. I applied log, square root and cube root transformations to both the explanatory and response variables, for the simple linear model, and to all variables for the multiple linear models, but this only made the variables less linear and affected the models' performances, so I opted to proceed with the dataset as it was, with the aim of using the cross validation to deal with this issue.

3 Linear Regression modelling

With any machine learning model, such as the linear regression model in this report, it is advisable to employ model validation methods to ensure that the model generalizes well to new, unseen data, which helps avoid overfitting or underfitting. Model validation also helps assess the model's predictive performance on the unseen datasets, ensuring that the relationships identified by the model on the training data are also identified in other datasets. There are a few methods of model validation, including splitting the original dataset into a training and testing split, cross validation (K-fold or Leave One Out method), bootstrapping, and examining the model performance metrics. These methods are generally applicable in most cases, but certain factors, such as the size of the dataset, the purpose of your analysis, and the computational resources available to you, can sometimes determine which methods you have to use. Since the dataset in this report is very small (only 41 observations), it would be better to use cross validation methods like k-fold instead of splitting the dataset, which would significantly reduce the data size for training the model.

Therefore, for my analysis in this report, I employed K-fold cross validation for both the simple and multiple linear regression models. This made sure that each data point, shuffled across different folds, is used for both training and validating the model, and also reduced the variance in performance estimation as it averages the performance across different folds and multiple repeats to give a more reliable performance measure. I defined a function, ‘cv_error’, that would create 10 folds from all the observations in the dataset to make sure we had enough training data. This function would then split the data into training and testing sets for all the folds, fit the linear model on the training split, perform prediction on the test split, and calculate the mean squared error (MSE) for each fold. I also define the function ‘repeated cv_error’ which would then repeat this process five times to get a range of possible MSE values, and then return the average of all MSE values. Getting the average of many repeats would make the results more reliable and provide a better assessment of each of the models’ predictive abilities.

3.1 Simple Linear Regression

For the simple linear model, I was looking for coefficients of the model that would fit the formula **y^ (y hat) = a + bx**, where **b** is the slope of the least squares line, and **a** is the intercept, i.e., where the least squares line crosses the y-axis, or the value of y when x is 0.

Following the exploratory analysis, I identified that the variable ‘manu’ had the highest correlation with the target variable, ‘SO_2 ’, and would therefore be the best predictor for the simple linear model. I created the simple linear model with ‘SO_2 ’ as the y variable and ‘manu’ as the x variable and plotted the model with the line of best fit as shown in Fig. 4 below.

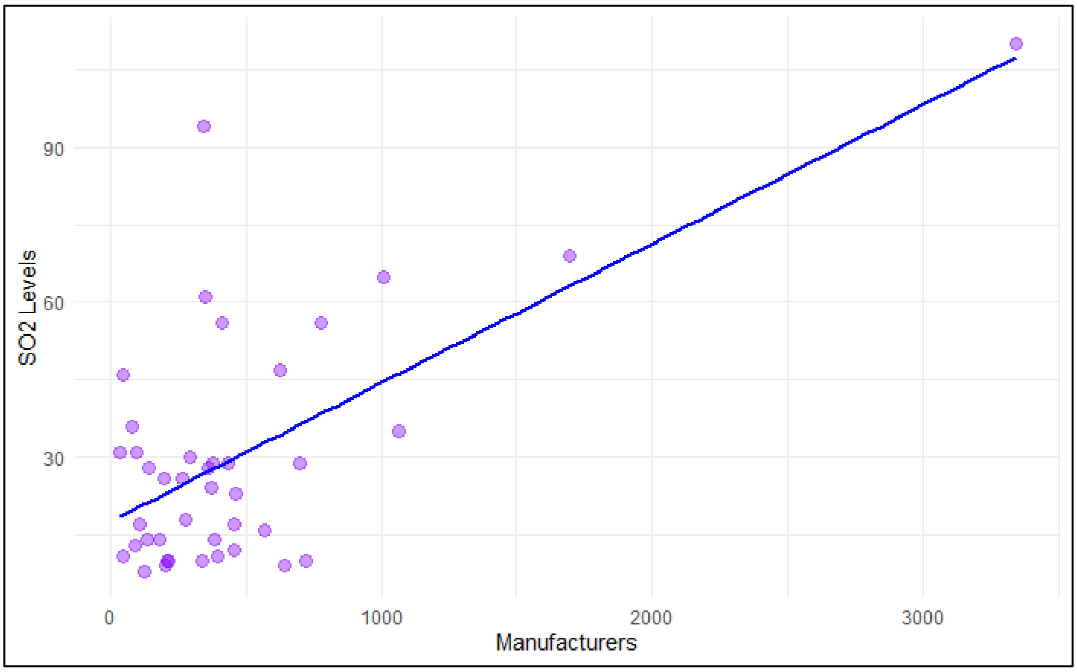


Fig. 4: Simple Linear model with least squares line

Results

From the summary and visual plot in Fig. 4, I could see that the equation of the least squares line was equal to **y^(y hat) = 17.61 + 0.03x**. In other words:

$$\mathbf{SO_2 = 17.61 + 0.03\ manu}$$

The slope of the line, b, is 0.03. This means that each additional manufacturing enterprise with 20 or more workers is predicted to increase the SO_2 content in the air by 0.03 micrograms per cubic meter. The positive coefficient for ‘manu’ also showed that there is a positive association between ‘manu’ and SO_2 levels. The intercept, a, is 17.61. This means that when the number of manufacturing enterprises is zero, the SO_2 content in the air is 17.61 micrograms per cubic meter.

The R-squared (R^2) value for the model was 0.4157, which means that around 42% of the variability in the SO_2 levels was explained by the predictor variable , ‘manu’. The F-statistic, 27.75 with p-value 5.36×10−6 indicates the model’s overall significance and this value shows that the model as a whole was a better fit than just modelling with the intercept alone. The five-number summary of the residuals showed some variability in prediction errors, with the comparatively large maximum residual value, 67.18, indicating probable outlier effect or some potentially large deviations that could affect the performance of the model.

Diagnostics checking

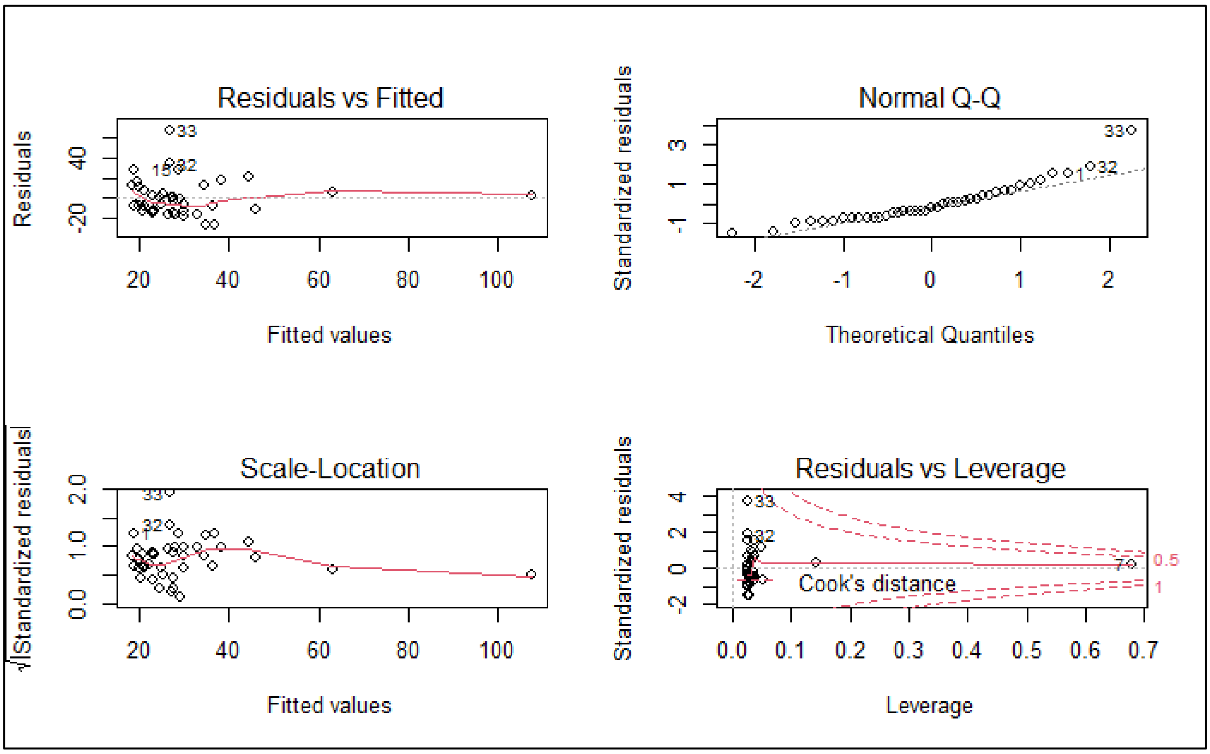


Fig. 5: Diagnostic plots for the Simple Linear Regression model

1. (TOP LEFT: Residuals vs Fitted): There seems to be a slightly upward trend, which might indicate that the variance of residuals is not constant, thus violating the homoscedasticity assumption.

2. (BOTTOM LEFT: Scale Location): There is a slightly upward trend, suggesting non-constant variance in the residual errors, thus violating the homoscedasticity assumption.

3. (TOP RIGHT: Normal Q-Q): Most the points fall approximately along this reference line, except a small minority of potential outliers on the top end, so we can assume normality.

4. (BOTTOM RIGHT: Residuals vs Leverage): The plot shows that there are a few points with relatively higher leverage, which could be potential influential outliers.

3.2 Multiple Linear Regression

I then conducted multiple linear regression analyses with 1) all subsets, 2) forward stepwise, and 3) backward stepwise selection methods to compare how well each one would perform and to identify which one could best predict the SO_2 content in the air given multiple explanatory variables.

3.2.1 All subsets

The goal for this model was to evaluate all possible combinations of explanatory variables to determine which combination had the best predictive performance. This method is typically more extensive as it exhausts all possible combinations to find the most optimal one for the prediction of the target variable. I then used the Mean Square Error (MSE) values calculated using the k-fold cross validation function to assess the predictive performance of each combination to determine how many predictor variables would have the best prediction of SO_2 content in the air. The MSE values for each combination of variables were as follows:

Number of predictors	MSE value
1	319.221
2	256.577
3	257.116
4	258.751
5	226.602
6	252.179

Table 1: All Subsets MSE values

As a general rule, the lower the MSE value, the better the model is at predicting the target variable. Therefore, from Table 1 above, the model with 5 predictors seemed to be the best model, as it has the lowest MSE value. The table also showed that the model with just a single predictor has the worst performance, which was in line with the MSE value for the simple linear regression model discussed later in the report.

3.2.2 Backward stepwise

I then proceeded to create a multiple linear model which starts off fitted with all possible predictors and removes them one by one using the AIC (Akaike Information Criteria) to inform the algorithm when to stop the removal of predictors. The AIC helps ensure there is a balance between model complexity and model fit by penalizing models with more predictor variables, thus reducing possible overfitting and prioritizing models with lower AIC values. The model started with all 6 numerical predictor variables, and using the AIC, was able to reduce the number of predictor variables to 5.

Results

From the summary of the model, I could see that the formula, **y^ (y hat) = 100.15 – 1.12x1 + 0.06x2 – 0.04x3 – 3.08x4 + 0.42x5**, was equal to:

$$\mathbf{SO_2 = 100.15 - 1.12\ temp + 0.06\ manu - 0.04\ popul - 3.08\ wind + 0.42\ precip}$$

- The intercept is 100.15, which means when all predictor variables are equal to zero, the SO_2 content in the air is 100.15 micrograms per cubic meter.
- For each Fahrenheit increase in the average annual temperature, the SO_2 content in the air decreases by 1.12 micrograms per cubic meter.
- The SO_2 content in the air increases by 0.06 micrograms per cubic meter for each additional manufacturing enterprise employing 20 or more workers.
- For each additional thousand people increase in population size, the SO_2 content in the air is predicted to decrease by 0.039 micrograms per cubic meter.
- For each additional mph increase in average annual wind speed, the SO_2 content in the air decreases by 3.08 micrograms per cubic meter. However, the relationship between the two variables is marginally significant as shown by the p-value, suggesting a weak association between them.
- Lastly, the SO_2 content in the air is predicted to decrease by 0.419 micrograms per cubic meter for each additional inch in the average annual precipitation.

The R-squared (R^2) value showed that the backward stepwise model was able to explain approximately 66.85% of the variability in the SO_2 content in the air. The final model includes the predictors temp, manu, popul, wind, and precip, with ‘manu’ showing the strongest positive association with SO_2. The F-statistic, 14.12 with p-value $1.41 \times 10{-}7$ also showed that the model was statistically significant and indicates that the chosen predictor variables explain a significant proportion of the variability in SO_2 content in the air.

Diagnostics checking

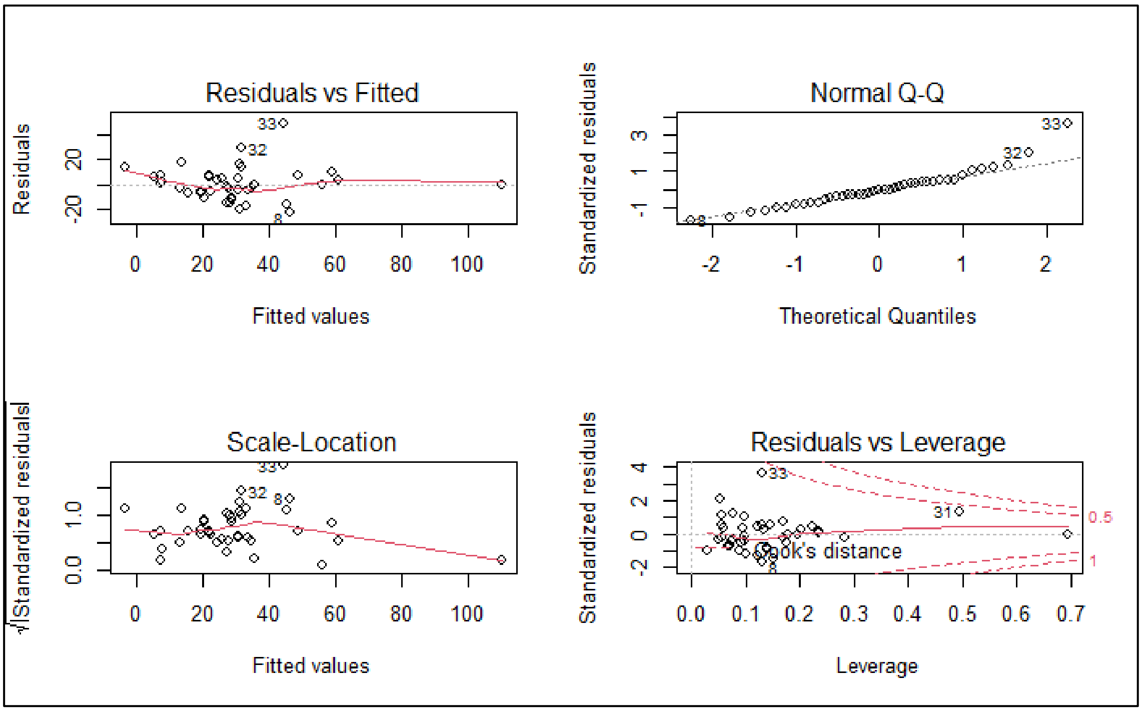


Fig. 6: Diagnostic plots for Backward stepwise Multiple Linear model

1. (TOP LEFT: Residuals vs Fitted): There seems to be a downward, then upward trend, which might indicate that the variance of residuals is not constant, thus violating the homoscedasticity assumption.

2. (BOTTOM LEFT: Scale Location): There seems to be a upward, then downward trend, suggesting non-constant variance in the residual errors, thus violating the homoscedasticity assumption.

3. (TOP RIGHT: Normal Q-Q): Most the points fall approximately along this reference line, except a small minority of potential outliers on the top end, so we can assume normality.

4. (BOTTOM RIGHT: Residuals vs Leverage): The plot shows that there are a few points with relatively higher leverage, which could be potential influential outliers.

3.2.3 Forward stepwise

I then created a multiple linear model which starts off fitted with only the intercept, then incrementally adds predictors one by one using the AIC (Akaike Information Criteria) to inform the algorithm when to stop adding predictors to the model. The model started with only the intercept, and using the AIC, was able to add 3 predictor variables to arrive at the final model.

Results

From the summary of the model, I could see that the formula, **y^ (y hat) = 6.97 + 0.07x$_1$ - 0.05x$_2$ + 0.16x$_3$**, was equal to:

SO_2 = 6.97 + 0.07 manu - 0.05 popul + 0.16 predays

- For each additional manufacturing enterprise employing 20 or more workers, the SO_2 content in the air increases by 0.07 micrograms per cubic meter.
- For each additional thousand people increase in population size, the SO_2 content in the air is predicted to decrease by 0.05 micrograms per cubic meter.
- Lastly, the SO_2 content in the air is predicted to increase by 0.16 micrograms per cubic meter for each additional day with precipitation annually.

The R-squared (R^2) value showed that the backward stepwise model was able to explain approximately 61.74% of the variability in the SO_2 content in the air. The final model includes the predictors manu, popul, and predays, with 'manu' showing the strongest positive association with SO_2. The F-statistic, 19.9 with p-value 7.54×10−8 also showed that the model was statistically significant and indicates that the chosen predictor variables explain a significant proportion of the variability in SO_2 content in the air.

3.3 Predictive ability

In order to compare the predictive abilities of the models discussed above, I used the 'cv_error' function I created earlier to create 10 folds from all the observations in the dataset, ensuring I had enough training data. This function would split the data into training and testing sets for all the folds, fit the linear model on the training split, perform prediction on the test split, and calculate the mean squared error (MSE) for each fold. For reliability of results, I then used the function 'repeated cv_error', which repeated this process five times to get a range of possible MSE values, and then return the average of all MSE values. The average MSE values of the models were as follows:

Model	MSE for single run	Average MSE for 5 runs
Simple Linear	348.513	330.879
Backward stepwise	242.549	239.482
Forward stepwise	241.113	247.772

Table 2: Mean Square Error values for all models

As mentioned earlier, the lower the MSE value, the better the model is at predicting the response variable given the list of predictor variables. Therefore, using the average MSE value for each model, I concluded that the multiple linear regression model with the backward stepwise method was the best predictor of SO_2 content in the air. The model was able to explain a significant amount of the variability in SO_2 values using 5 predictor variables – temp, manu, popul, wind, precip.

4 Discussion

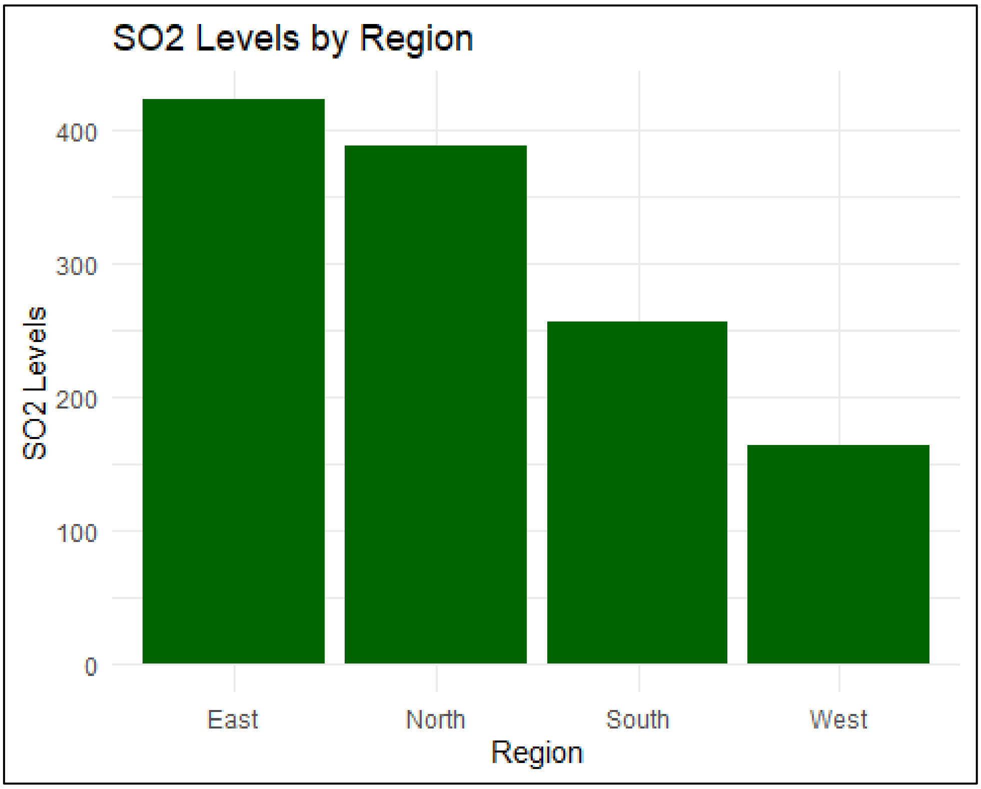
This study was able to provide insight into some of the key factors that affect the amount of SO_2 content in the air. The linear regression models used in the analyses were able to identify the main factors that influence the SO_2 levels, with the number of manufacturing enterprises with more than twenty workers emerging as a primary contributor, suggesting that the increase in industrial activity in a region coincides with increased levels of air pollution. Since I chose the backward stepwise model, this suggests that the number of manufacturing factories, the population size, and the number of days with precipitation annually have the most significant influence on the amount of SO_2 in the air, whether negative or positive.

There were, however, a few limitations of the study, including the small dataset, which limited the statistical power of the regression models and possibly affected how well the resulting model would generalize to new, unseen data. Additionally, the non-linearity of relationships between predictors and the target variable also affected the accuracy of the analysis. For future analyses, selecting a larger dataset would help to account for some of these issues, and if lack of normality is still an issue, consideration could be taken to use nonlinear models that are more robust to nonlinearity in data.

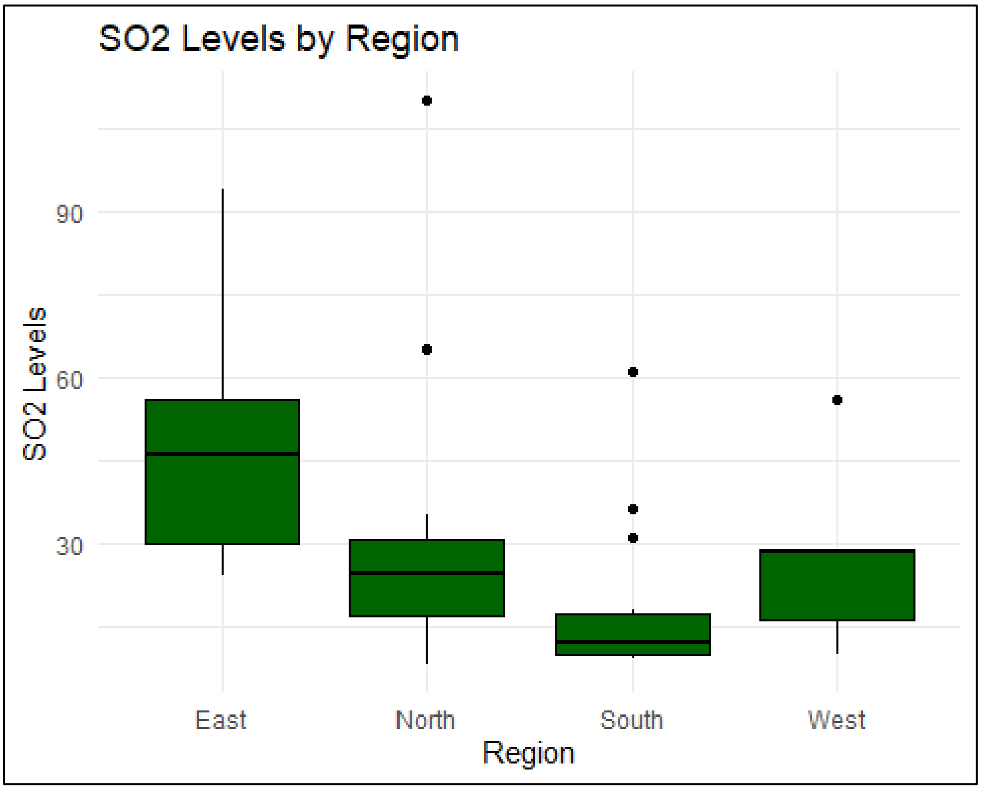
In conclusion, despite the limitations of the study, it still provides valuable insight into how different ecological and industrial factors affect the amount of SO_2 in the air and in turn affect air pollution, and it can be used to inform environmental scientists, governments, etc., on how to set in place regulations that would help curb the increase in air pollution. It also provides a guideline for which machine learning methods would be better suited for the purposes of predicting the amount of air pollution given a list of similar factors.

5 Appendices (R Code)

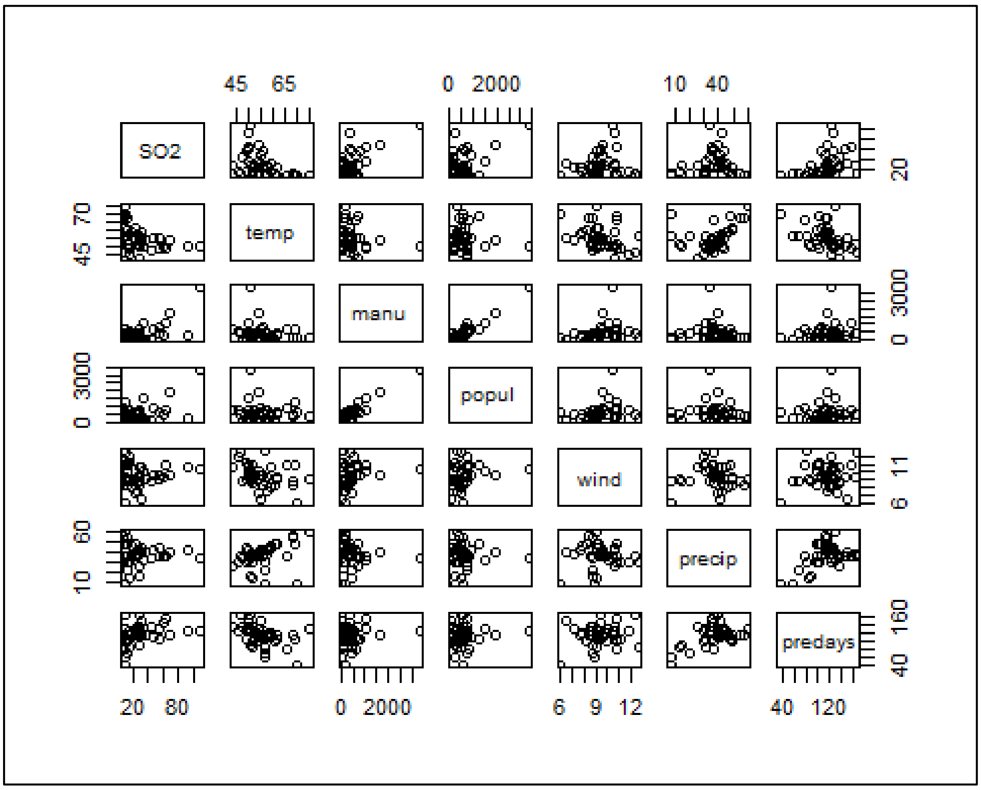
```
knitr::opts_chunk$set(echo = TRUE)
# importing the necessary libraries
library(dplyr)
library(Hmisc)
library(ggplot2)
## opening the datasets
pollution <- read.csv("Pollution.csv")
View(pollution)
## getting a glimpse of the dataset
head(pollution)
## five-number summary
summary(pollution)
## checking for missing values
missing <- colSums(is.na(pollution))
missing
## histograms for SO2 levels by region
ggplot(pollution, aes(x = factor(region), y = SO2)) +
  geom_col(fill = "darkgreen") +
  labs(title = "SO₂ Levels by Region", x = "Region", y = "SO₂ Levels") +
  theme_minimal()
```



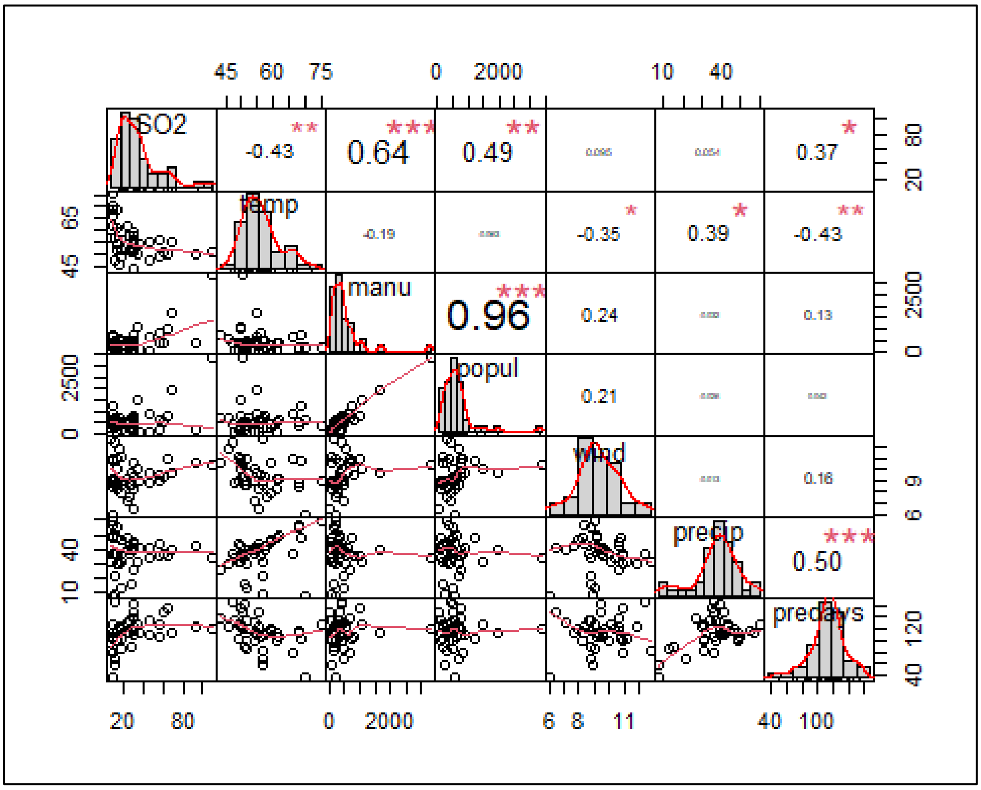
```
# boxplot to see the spread, median and potential outliers
ggplot(pollution, aes(x = factor(region), y = SO2)) +
  geom_boxplot(fill = "darkgreen", color = "black") +
  labs(title = "SO₂ Levels by Region", x = "Region", y = "SO₂ Levels") +
  theme_minimal()
```



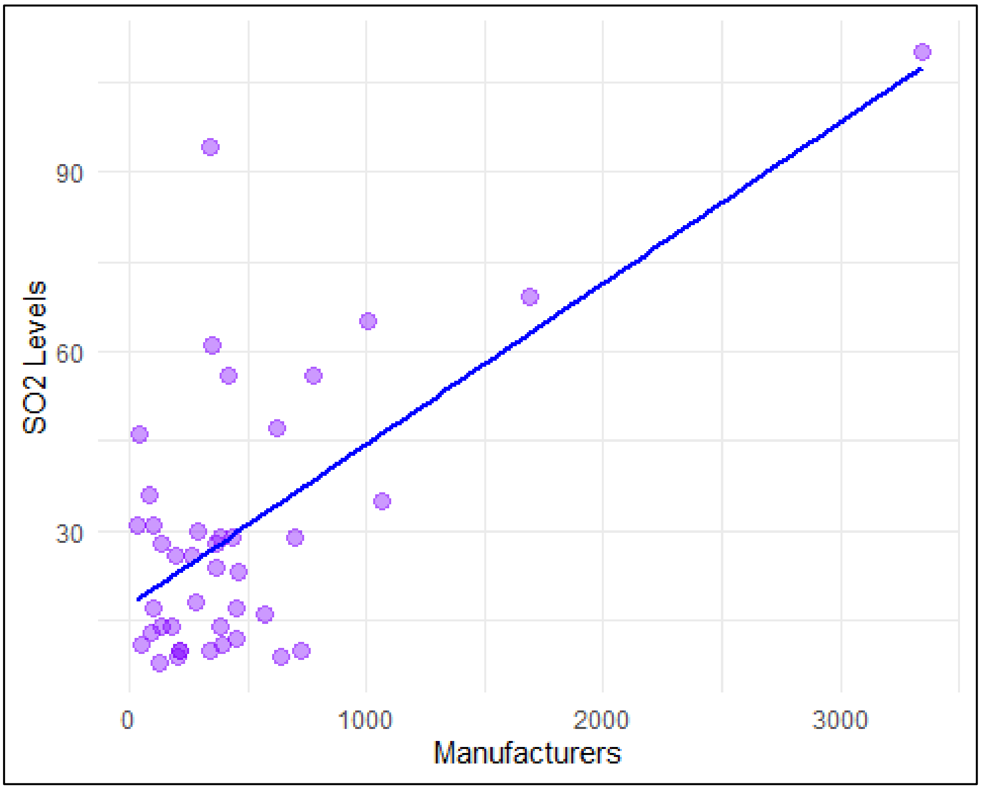
```
## checking the correlation between numeric variables
num_pollution <- pollution[,c(-8)]
pairs(num_pollution) #shows the association between variables
```



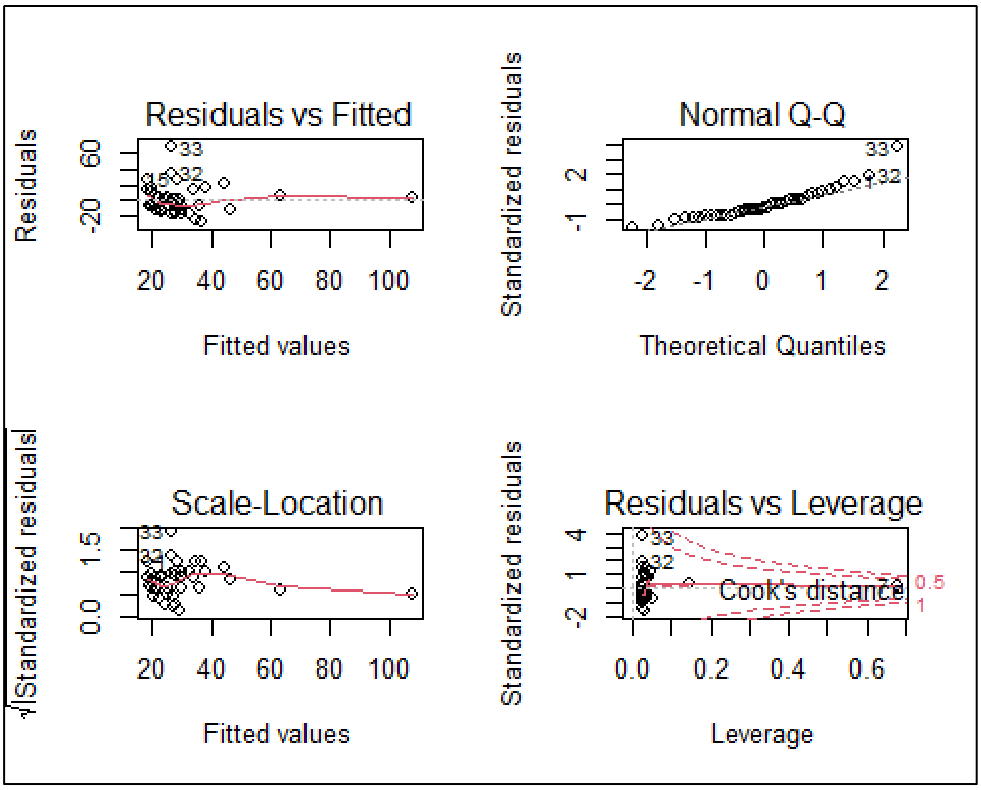
```
# corrplot
library(PerformanceAnalytics)  ## install this
chart.Correlation(num_pollution)
```



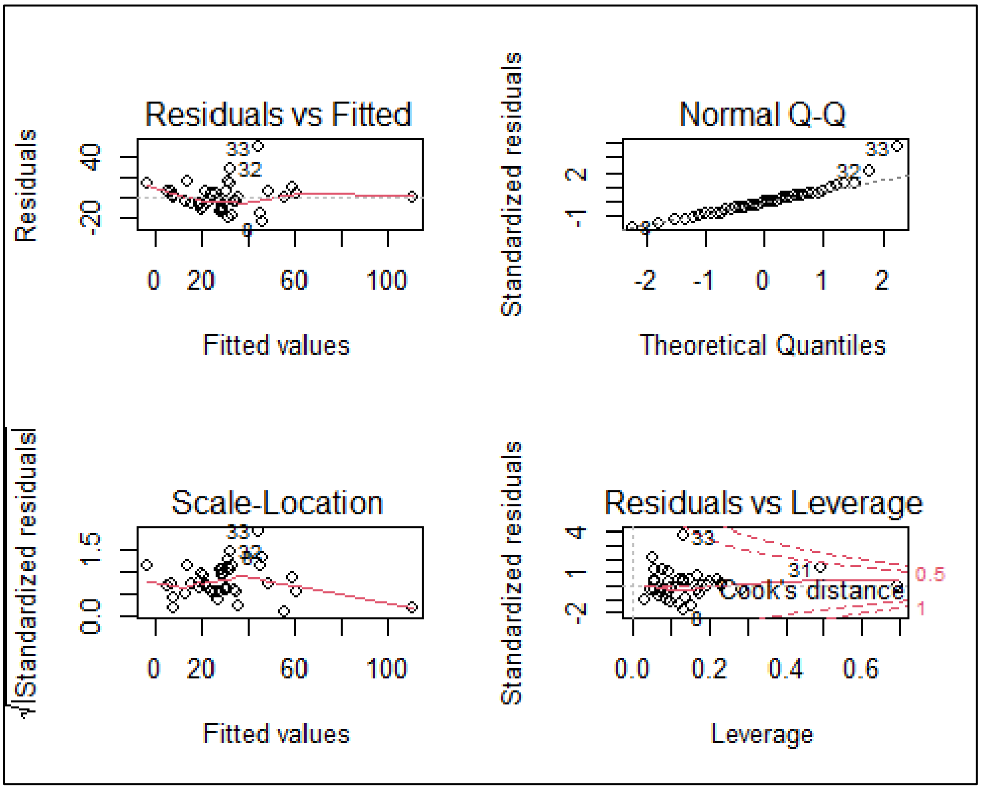
```
# Sample data: Assuming your data frame is called num_pollution
# Apply the function to each numeric column
outlier_counts <- sapply(num_pollution, function(x) {
  if (is.numeric(x)) {
    Q1 <- quantile(x, 0.25, na.rm = TRUE)
    Q3 <- quantile(x, 0.75, na.rm = TRUE)
    IQR <- Q3 - Q1
    lower_bound <- Q1 - 1.5 * IQR
    upper_bound <- Q3 + 1.5 * IQR
    sum(x < lower_bound | x > upper_bound)  # Count outliers
  } else {
    NA  # Skip non-numeric columns
  }
})

# Remove any NA values from the result
outlier_counts <- outlier_counts[!is.na(outlier_counts)]

# Display the count of outliers for each variable
print(outlier_counts)

# Refined k-fold cross-validation function
cv_error <- function(formula, data, k = 10) {
  #set.seed(123) #for reproducibility
  n <- nrow(data)
  folds <- sample(rep(1:k, length.out = n))  # Create folds
  errors <- numeric(k)  # To store MSE for each fold

  for (i in 1:k) {
    # Split data into training and testing sets based on fold
    test_index <- which(folds == i)
    train_data <- data[-test_index, ]
    test_data <- data[test_index, ]

    # Fit the model on the training data
    model <- lm(formula, data = train_data)

    # Predict on the test data
    predictions <- predict(model, newdata = test_data)

    # Calculate mean squared error for this fold
    errors[i] <- mean((test_data$SO2 - predictions)^2)
  }

  # Return average cross-validated MSE
  mean(errors)
}

# New function to perform repeated k-fold cross-validation and average the MSE
repeated_cv_error <- function(formula, data, k = 10, repeats = 5) {
  set.seed(123)  # For reproducibility across multiple rounds
  mse_values <- numeric(repeats)

  for (i in 1:repeats) {
    mse_values[i] <- cv_error(formula, data, k)
  }

  mean(mse_values)  # Return the average MSE over all repeats
}

simple_model <- lm(SO2~manu, data=num_pollution)
summary(simple_model)
# Run k-fold cross-validation on a simple linear model with SO2 ~ manu
simple_mse <- cv_error(SO2 ~ manu, data = num_pollution, k = 10)
print(paste("Cross-validated MSE for Simple Linear Regression:", round(simple_mse, 3)))

# getting the average MSE for 10 folds and 5 repeats
simple_avgmse <- repeated_cv_error(SO2~manu, data = num_pollution, k = 10, repeats = 5)
print(paste("Average MSE for Simple Model:", round(simple_avgmse, 3)))
# loading the necessary pakckages for plotting SLR line of best fit
library(ggplot2)
library(ggpubr)

# Plot the data using ggplot2 and include the regression line with the equation
ggplot(num_pollution, aes(x = manu, y = SO2)) +
  geom_point(color = rgb(0.5, 0, 1, alpha = 0.4), size = 3) +  # Scatter plot
  geom_smooth(method = "lm", se = FALSE, color = "blue", linetype = "solid") +  # Regression line
  labs(x = "Manufacturers", y = "SO2 Levels") +  # Axis labels
  theme_minimal()  # for a cleaner look

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



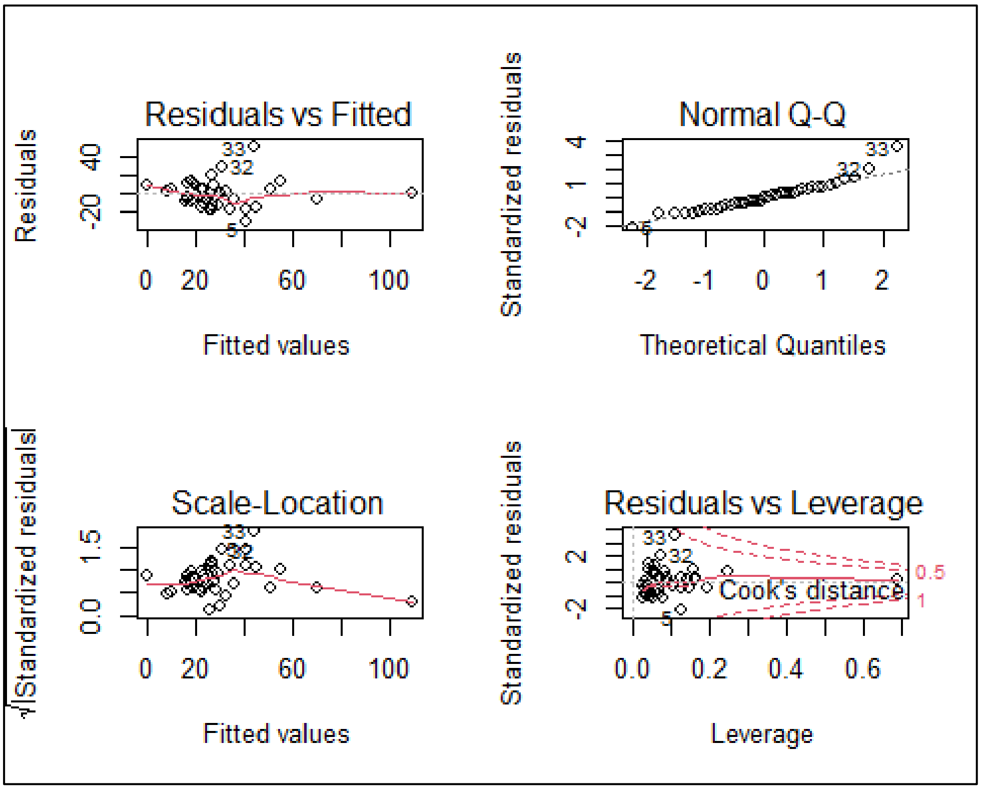
```r
# diagnostics checking
par(mfrow = c(2, 2))  # Set up for 4 plots
plot(simple_model)
```



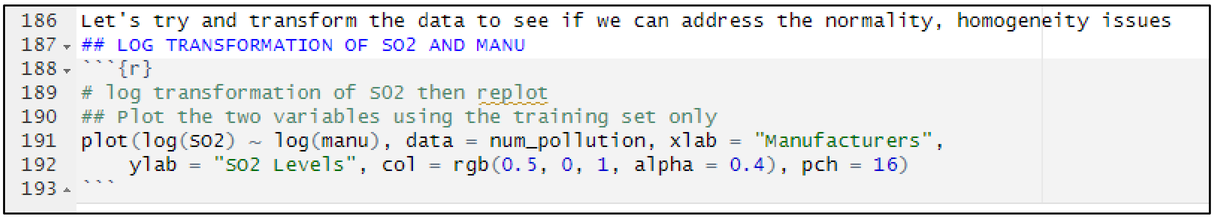
```r
# load the package
library(leaps)

# Fit all possible models
all_subsets <- regsubsets(SO2 ~ temp + manu + popul + wind + precip + pred
ays, data = num_pollution, nvmax = 6)
#summary(all_subsets)
# loading required package
if (!requireNamespace("leaps", quietly = TRUE)) install.packages("leaps")
library(leaps)

# Cross-validate each model in subset selection
mse_values <- sapply(1:6, function(i) {
  # Build formula dynamically with the best predictors of the subset
  predictors <- names(coef(all_subsets, i))

  # Exclude intercept from the predictors
  if (length(predictors) > 1) {
    formula <- as.formula(paste("SO2 ~", paste(predictors[-1], collapse = 
" + ")))
  } else {
    formula <- as.formula("SO2 ~ 1")  # Only intercept model
  }

  # Perform cross-validation and get MSE
  cv_error(formula, data = num_pollution, k = 10)
})

# Display cross-validated MSE for each model
mse_results <- data.frame(Num_Predictors = 1:6, MSE = round(mse_values, 3)
)
print(mse_results)

# Fit the full model with all predictors
full_model <- lm(SO2 ~ temp + manu + popul + wind + precip + predays, data
= num_pollution)

# Perform backward selection
backward_model <- step(full_model, direction = "backward")

# Cross-validate the backward selection model
backward_mse <- cv_error(formula(backward_model), data = num_pollution, k
= 10)
print(paste("Cross-validated MSE for Backward Selection Model:", round(bac
kward_mse, 3)))

# getting the average MSE for 10 folds and 5 repeats
bwd_avgmse <- repeated_cv_error(formula(backward_model), data = num_pollut
ion, k = 10, repeats = 5)
print(paste("Average MSE for Backward Model:", round(bwd_avgmse, 3)))
# bw model coefficients
summary(backward_model)
# diagnostics checking
par(mfrow = c(2, 2))  # Set up for 4 plots
plot(backward_model)
```



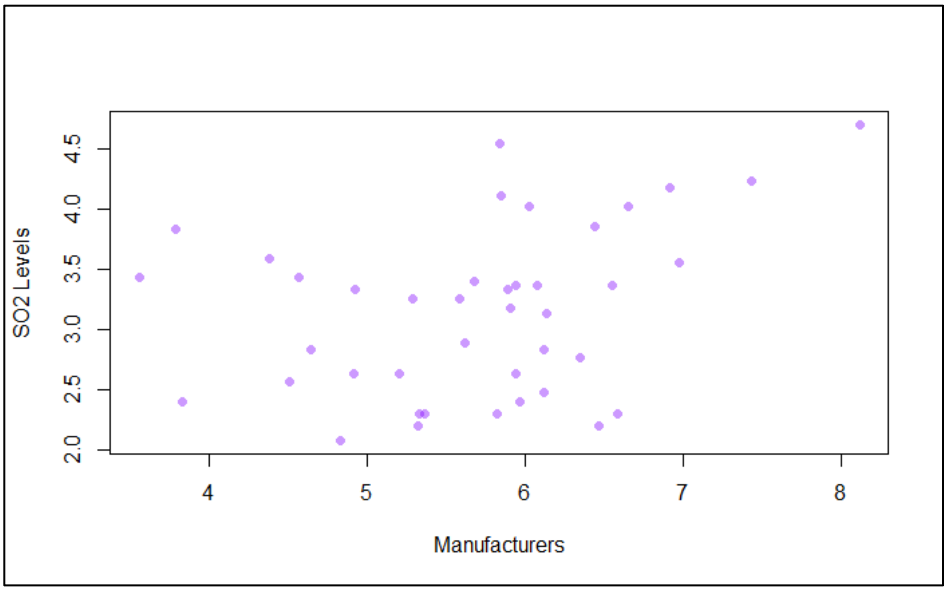
```
# Fit a minimal model (intercept only)
minimal_model <- lm(SO2 ~ 1, data = num_pollution)

# Perform forward selection
forward_model <- step(minimal_model, direction = "forward", scope = formula(full_model))

# Cross-validate the forward selection model
forward_mse <- cv_error(formula(forward_model), data = num_pollution, k = 10)
print(paste("Cross-validated MSE for Forward Selection Model:", round(forward_mse, 3)))

# getting the average MSE for 10 folds and 5 repeats
fwd_avgmse <- repeated_cv_error(formula(forward_model), data = num_pollution, k = 10, repeats = 5)
print(paste("Average MSE for Forward Model:", round(fwd_avgmse, 3)))
# fw model coefficients
summary(forward_model)
# diagnostics checking
par(mfrow = c(2, 2))  # Set up for 4 plots
plot(forward_model)
```

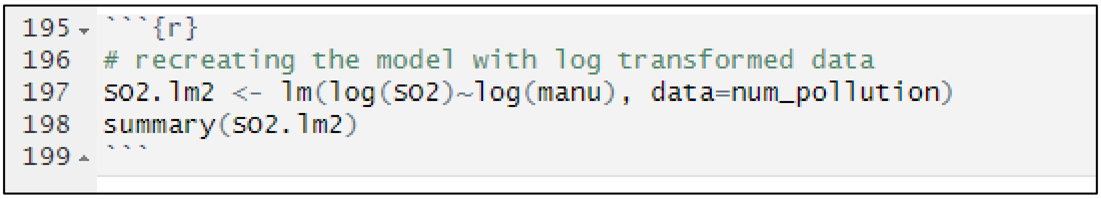


<u>Failed attempt with transformation (sqrt and cube root transformations performed just as poorly).</u>

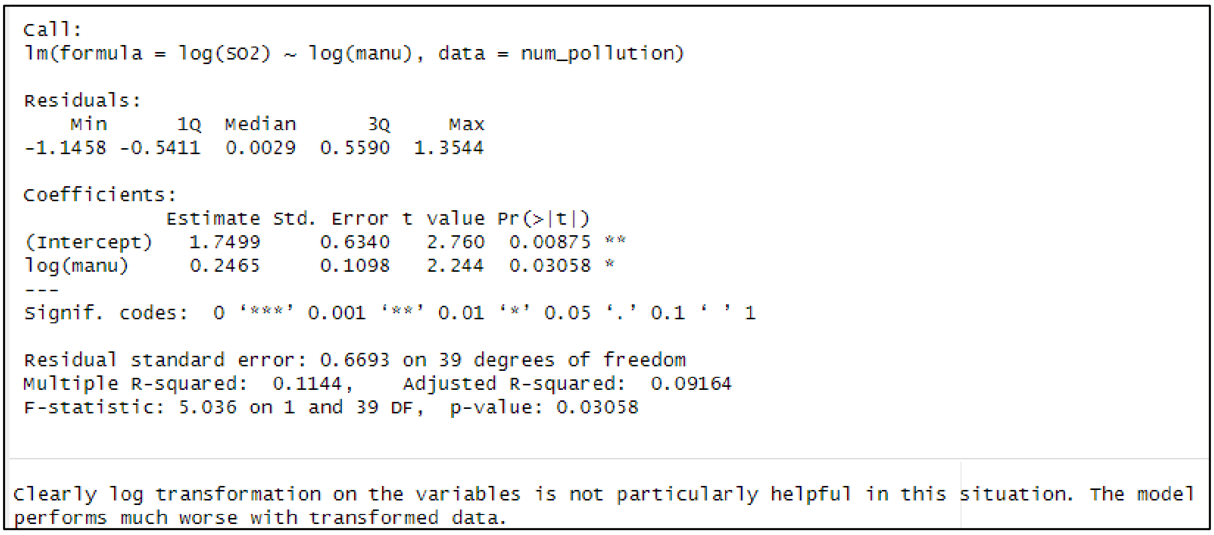


~~~
Let's try and transform the data to see if we can address the normality, homogeneity issues
## LOG TRANSFORMATION OF SO2 AND MANU
```{r}
# log transformation of SO2 then replot
## Plot the two variables using the training set only
plot(log(SO2) ~ log(manu), data = num_pollution, xlab = "Manufacturers",
     ylab = "SO2 Levels", col = rgb(0.5, 0, 1, alpha = 0.4), pch = 16)
```
~~~





~~~
```{r}
# recreating the model with log transformed data
so2.lm2 <- lm(log(so2)~log(manu), data=num_pollution)
summary(so2.lm2)
```
~~~



```
Call:
lm(formula = log(so2) ~ log(manu), data = num_pollution)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.1458 -0.5411  0.0029  0.5590  1.3544 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)   1.7499     0.6340   2.760  0.00875 **
log(manu)     0.2465     0.1098   2.244  0.03058 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6693 on 39 degrees of freedom
Multiple R-squared:  0.1144,	Adjusted R-squared:  0.09164 
F-statistic: 5.036 on 1 and 39 DF,  p-value: 0.03058
```

Clearly log transformation on the variables is not particularly helpful in this situation. The model performs much worse with transformed data.