STAT1006 Final project

Bruce Odek Omondi

2024-09-19

## R Markdown

# importing the necessary libraries   
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(Hmisc)

##   
## Attaching package: 'Hmisc'

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

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

library(ggplot2)

## opening the datasets  
pollution <- read.csv("Pollution.csv")  
View(pollution)

## getting a glimpse of the dataset   
head(pollution)

## SO2 temp manu popul wind precip predays region  
## 1 46 47.6 44 116 8.8 33.36 135 East  
## 2 11 56.8 46 244 8.9 7.77 58 South  
## 3 24 61.5 368 497 9.1 48.34 115 East  
## 4 47 55.0 625 905 9.6 41.31 111 East  
## 5 11 47.1 391 463 12.4 36.11 166 South  
## 6 31 55.2 35 71 6.5 40.75 148 South

All the variables are numerical, save for ‘region’ which contains categorical data.

## five-number summary   
summary(pollution)

## SO2 temp manu popul   
## Min. : 8.00 Min. :43.50 Min. : 35.0 Min. : 71.0   
## 1st Qu.: 13.00 1st Qu.:50.60 1st Qu.: 181.0 1st Qu.: 299.0   
## Median : 26.00 Median :54.60 Median : 347.0 Median : 515.0   
## Mean : 30.05 Mean :55.76 Mean : 463.1 Mean : 608.6   
## 3rd Qu.: 35.00 3rd Qu.:59.30 3rd Qu.: 462.0 3rd Qu.: 717.0   
## Max. :110.00 Max. :75.50 Max. :3344.0 Max. :3369.0   
## wind precip predays region   
## Min. : 6.000 Min. : 7.05 Min. : 36.0 Length:41   
## 1st Qu.: 8.700 1st Qu.:30.96 1st Qu.:103.0 Class :character   
## Median : 9.300 Median :38.74 Median :115.0 Mode :character   
## Mean : 9.444 Mean :36.77 Mean :113.9   
## 3rd Qu.:10.600 3rd Qu.:43.11 3rd Qu.:128.0   
## Max. :12.700 Max. :59.80 Max. :166.0

This gives us information about the location(median), spread(quartiles), and range(min and max values) of values in each variable.

## checking for missing values  
missing <- colSums(is.na(pollution))  
missing

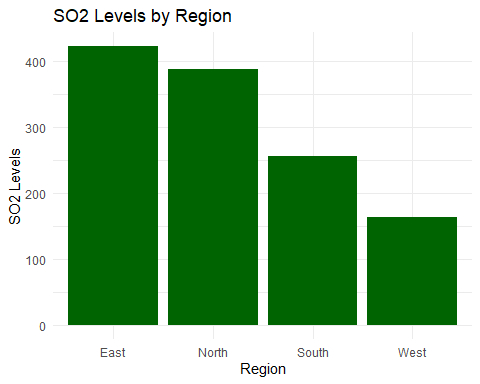
## SO2 temp manu popul wind precip predays region   
## 0 0 0 0 0 0 0 0

Clearly, none of the variables have missing values.

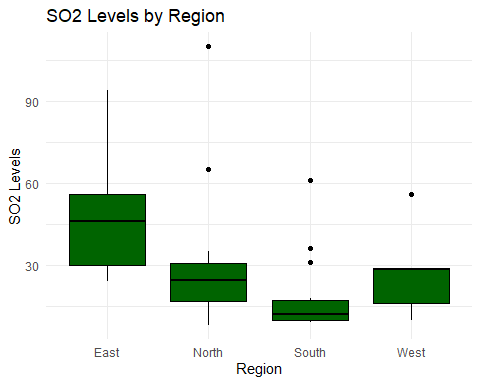
# Exploratory Data Analysis

I was interested in finding out how the variables vary by region, seeing as it is the only categorical variable in the dataset. To do this, I plotted a histogram of the other variables against the region.

## 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()

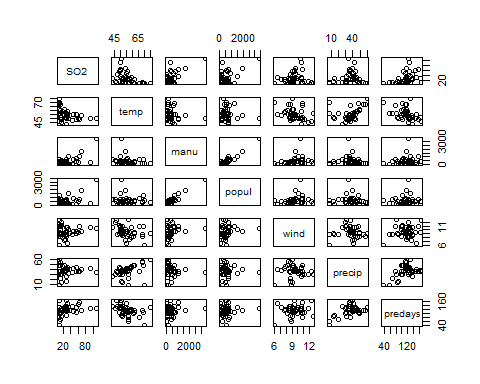


# boxplot to see the spread, median and potential outliers   
ggplot(pollution, aes(x = factor(region), y = SO2)) +  
 geom\_boxplot(fill = "darkgreen", color = "black") +  
 labs(title = "SO₂ Levels by Region", x = "Region", y = "SO₂ Levels") +  
 theme\_minimal()



Let’s now check the association and correlation between the variables

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



# corrplot   
library(PerformanceAnalytics) ## install this

## Loading required package: xts

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

##   
## ######################### Warning from 'xts' package ##########################  
## # #  
## # The dplyr lag() function breaks how base R's lag() function is supposed to #  
## # work, which breaks lag(my\_xts). Calls to lag(my\_xts) that you type or #  
## # source() into this session won't work correctly. #  
## # #  
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #  
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop #  
## # dplyr from breaking base R's lag() function. #  
## # #  
## # Code in packages is not affected. It's protected by R's namespace mechanism #  
## # Set `options(xts.warn\_dplyr\_breaks\_lag = FALSE)` to suppress this warning. #  
## # #  
## ###############################################################################

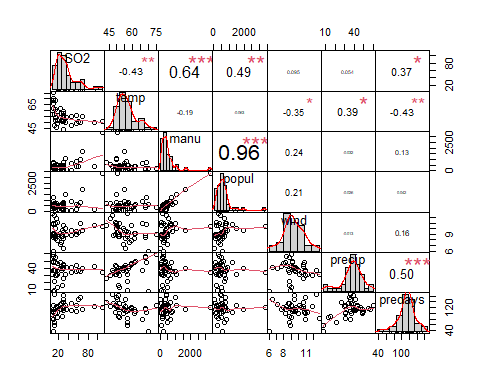
##   
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':  
##   
## first, last

##   
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':  
##   
## legend

chart.Correlation(num\_pollution)

 From the correlation matrix, we can see that the variable with the highest correlation to the response variable(SO2) is manu, i.e., the number of manufacturing enterprises employing 20 or more workers. This makes sense contextually as well, so it seems like the right variable for the Simple Linear Regression Model.

# checking for outliers

# 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)

## SO2 temp manu popul wind precip predays   
## 3 1 4 3 0 2 3

Since the dataset is very small, I manually checked each of the outliers in each variable and ascertained that they are genuine datapoints, they just happen to be very large compared to the other observations. Therefore, I will not be dropping them. However, I will check if transformation can help alleviate their influence on the model(s).

Given that the dataset is very small, I opted for k-fold cross validation which would use the entire dataset instead of my initial plan which was to split the data into training and testing (70/30) splits for cross validation. Using training and test splits would have greatly reduced the amount of data used for training the model and would affect model accuracy.

# MODELLING WITH CROSS VALIDATION

Let’s create the function for k-fold cross validation first

# 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  
}

# SLR

Create the SLR model with SO2 as the response and manu as the explanatory variable.

simple\_model <- lm(SO2~manu, data=num\_pollution)  
summary(simple\_model)

##   
## Call:  
## lm(formula = SO2 ~ manu, data = num\_pollution)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -26.976 -12.968 -3.495 6.710 67.177   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17.610574 3.691587 4.770 2.58e-05 \*\*\*  
## manu 0.026859 0.005099 5.268 5.36e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.17 on 39 degrees of freedom  
## Multiple R-squared: 0.4157, Adjusted R-squared: 0.4007   
## F-statistic: 27.75 on 1 and 39 DF, p-value: 5.363e-06

# 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)))

## [1] "Cross-validated MSE for Simple Linear Regression: 330.142"

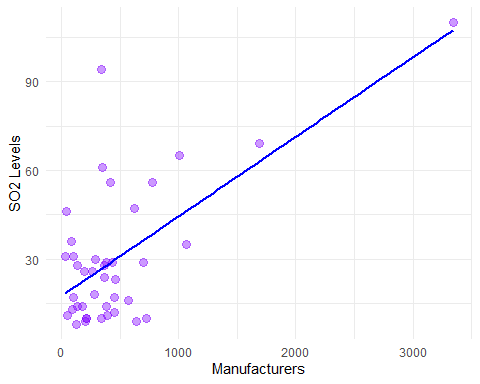
# getting the average MSE for 10 folds and 5 repeats  
simple\_avgmse <- repeated\_cv\_error(SO2~manu, data = num\_pollution, k = 10, repeats = 5)  
print(paste("Average MSE for Simple Model:", round(simple\_avgmse, 3)))

## [1] "Average MSE for Simple Model: 330.879"

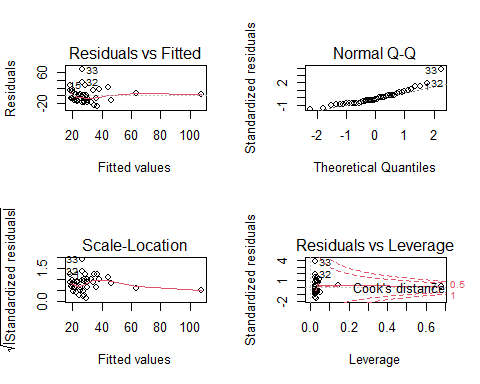
I ran the 10 folds for 5 repeats to get an average MSE value, which would account for variability of the different folds and runs.

# 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'



# diagnostics checking  
par(mfrow = c(2, 2)) # Set up for 4 plots  
plot(simple\_model)



# MLR (all subsets)

Start with the all subsets model

# load the package  
library(leaps)  
  
# Fit all possible models  
all\_subsets <- regsubsets(SO2 ~ temp + manu + popul + wind + precip + predays, 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)

## Num\_Predictors MSE  
## 1 1 319.221  
## 2 2 256.577  
## 3 3 257.116  
## 4 4 258.751  
## 5 5 226.602  
## 6 6 252.179

# MLR (Backward stepwise)

# 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")

## Start: AIC=226.37  
## SO2 ~ temp + manu + popul + wind + precip + predays  
##   
## Df Sum of Sq RSS AIC  
## - predays 1 22.1 7305.4 224.50  
## <none> 7283.3 226.37  
## - precip 1 427.3 7710.6 226.71  
## - wind 1 658.1 7941.4 227.92  
## - temp 1 892.5 8175.8 229.11  
## - popul 1 1443.1 8726.3 231.78  
## - manu 1 3640.1 10923.4 240.99  
##   
## Step: AIC=224.49  
## SO2 ~ temp + manu + popul + wind + precip  
##   
## Df Sum of Sq RSS AIC  
## <none> 7305.4 224.50  
## - wind 1 636.1 7941.5 225.92  
## - precip 1 785.4 8090.8 226.68  
## - popul 1 1447.5 8752.9 229.91  
## - temp 1 1517.4 8822.8 230.23  
## - manu 1 3636.8 10942.1 239.06

# 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(backward\_mse, 3)))

## [1] "Cross-validated MSE for Backward Selection Model: 242.549"

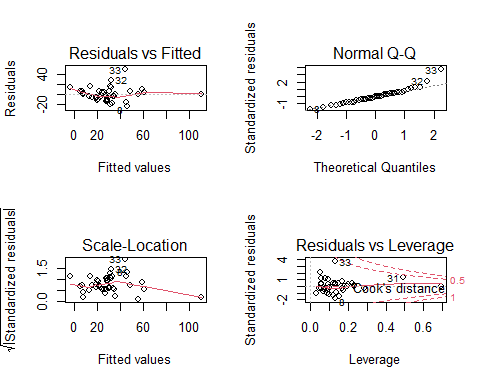
# getting the average MSE for 10 folds and 5 repeats  
bwd\_avgmse <- repeated\_cv\_error(formula(backward\_model), data = num\_pollution, k = 10, repeats = 5)  
print(paste("Average MSE for Backward Model:", round(bwd\_avgmse, 3)))

## [1] "Average MSE for Backward Model: 239.482"

# bw model coefficients  
summary(backward\_model)

##   
## Call:  
## lm(formula = SO2 ~ temp + manu + popul + wind + precip, data = num\_pollution)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -23.253 -7.655 -0.581 6.059 49.438   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 100.15245 30.27521 3.308 0.002182 \*\*   
## temp -1.12129 0.41586 -2.696 0.010707 \*   
## manu 0.06489 0.01554 4.174 0.000188 \*\*\*  
## popul -0.03933 0.01494 -2.633 0.012499 \*   
## wind -3.08240 1.76562 -1.746 0.089622 .   
## precip 0.41947 0.21624 1.940 0.060498 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14.45 on 35 degrees of freedom  
## Multiple R-squared: 0.6685, Adjusted R-squared: 0.6212   
## F-statistic: 14.12 on 5 and 35 DF, p-value: 1.409e-07

# diagnostics checking  
par(mfrow = c(2, 2)) # Set up for 4 plots  
plot(backward\_model)



# MLR (Forward stepwise)

# 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))

## Start: AIC=259.76  
## SO2 ~ 1  
##   
## Df Sum of Sq RSS AIC  
## + manu 1 9161.7 12876 239.73  
## + popul 1 5373.2 16665 250.31  
## + temp 1 4143.3 17895 253.23  
## + predays 1 3009.9 19028 255.74  
## <none> 22038 259.76  
## + wind 1 197.6 21840 261.40  
## + precip 1 65.0 21973 261.64  
##   
## Step: AIC=239.73  
## SO2 ~ manu  
##   
## Df Sum of Sq RSS AIC  
## + popul 1 3759.5 9116.6 227.58  
## + temp 1 2212.3 10663.8 234.00  
## + predays 1 1816.1 11060.0 235.50  
## <none> 12876.2 239.73  
## + precip 1 124.7 12751.4 241.33  
## + wind 1 80.6 12795.6 241.47  
##   
## Step: AIC=227.58  
## SO2 ~ manu + popul  
##   
## Df Sum of Sq RSS AIC  
## + predays 1 684.97 8431.7 226.37  
## + temp 1 577.98 8538.7 226.89  
## <none> 9116.6 227.58  
## + precip 1 148.26 8968.4 228.90  
## + wind 1 146.93 8969.7 228.91  
##   
## Step: AIC=226.37  
## SO2 ~ manu + popul + predays  
##   
## Df Sum of Sq RSS AIC  
## <none> 8431.7 226.37  
## + temp 1 257.679 8174.0 227.10  
## + wind 1 244.072 8187.6 227.17  
## + precip 1 4.262 8427.4 228.35

# 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)))

## [1] "Cross-validated MSE for Forward Selection Model: 241.113"

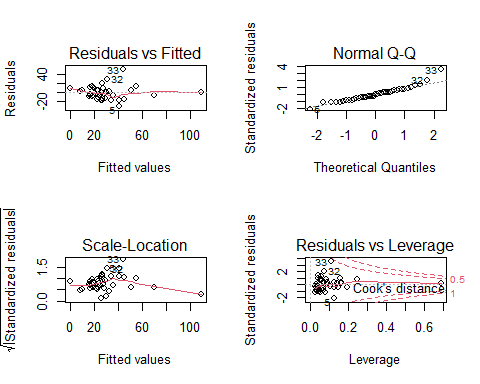
# getting the average MSE for 10 folds and 5 repeats  
fwd\_avgmse <- repeated\_cv\_error(formula(forward\_model), data = num\_pollution, k = 10, repeats = 5)  
print(paste("Average MSE for Forward Model:", round(fwd\_avgmse, 3)))

## [1] "Average MSE for Forward Model: 247.772"

# fw model coefficients   
summary(forward\_model)

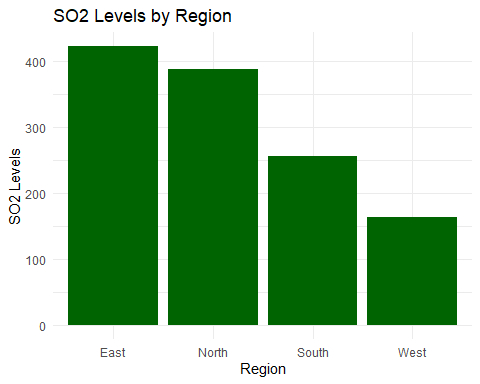
##   
## Call:  
## lm(formula = SO2 ~ manu + popul + predays, data = num\_pollution)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -29.445 -8.452 0.258 8.302 49.834   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.96585 11.77691 0.591 0.55779   
## manu 0.07433 0.01507 4.934 1.73e-05 \*\*\*  
## popul -0.04939 0.01454 -3.396 0.00165 \*\*   
## predays 0.16436 0.09480 1.734 0.09129 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.1 on 37 degrees of freedom  
## Multiple R-squared: 0.6174, Adjusted R-squared: 0.5864   
## F-statistic: 19.9 on 3 and 37 DF, p-value: 7.542e-08

# diagnostics checking  
par(mfrow = c(2, 2)) # Set up for 4 plots  
plot(forward\_model)

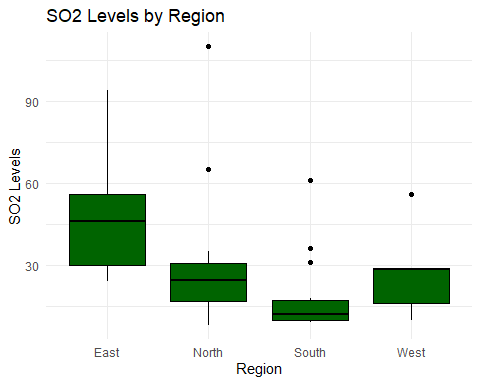
 Model diagnostic plots

# Appendix: All code for this report

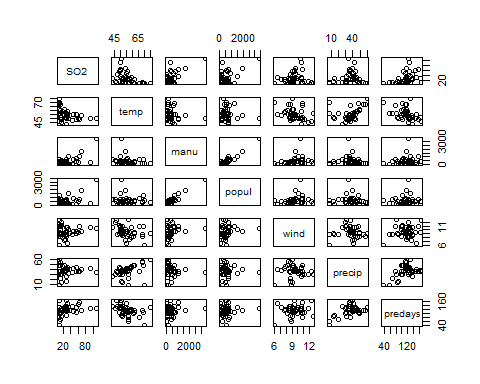
knitr::opts\_chunk$set(echo = TRUE)  
# importing the necessary libraries   
library(dplyr)  
library(Hmisc)  
library(ggplot2)  
## opening the datasets  
pollution <- read.csv("Pollution.csv")  
View(pollution)  
## getting a glimpse of the dataset   
head(pollution)  
## five-number summary   
summary(pollution)  
## 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()



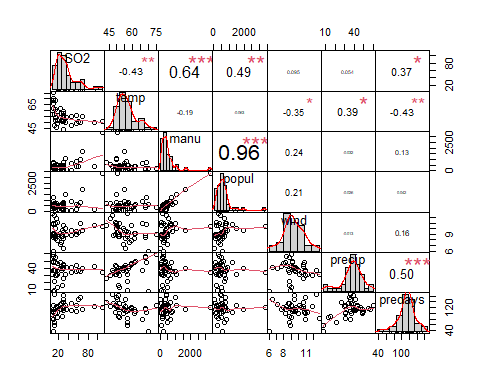
# boxplot to see the spread, median and potential outliers   
ggplot(pollution, aes(x = factor(region), y = SO2)) +  
 geom\_boxplot(fill = "darkgreen", color = "black") +  
 labs(title = "SO₂ Levels by Region", x = "Region", y = "SO₂ Levels") +  
 theme\_minimal()



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

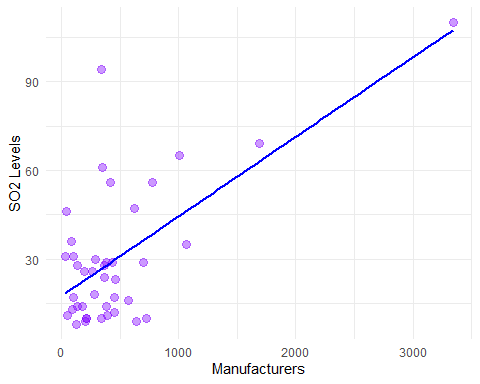


# corrplot   
library(PerformanceAnalytics) ## install this  
chart.Correlation(num\_pollution)

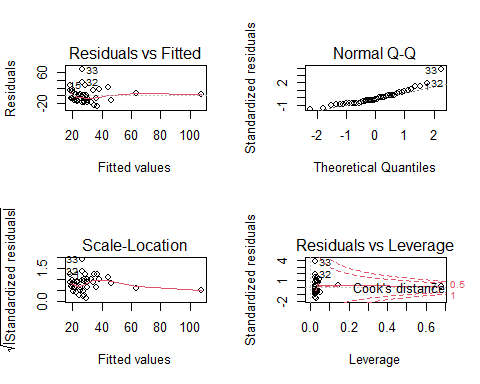


# Sample data: Assuming your data frame is called num\_pollution  
# Apply the function to each numeric column  
outlier\_counts <- sapply(num\_pollution, function(x) {  
 if (is.numeric(x)) {  
 Q1 <- quantile(x, 0.25, na.rm = TRUE)  
 Q3 <- quantile(x, 0.75, na.rm = TRUE)  
 IQR <- Q3 - Q1  
 lower\_bound <- 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

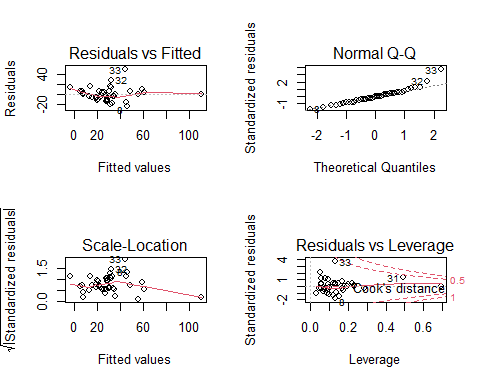
## `geom\_smooth()` using formula = 'y ~ x'



# diagnostics checking  
par(mfrow = c(2, 2)) # Set up for 4 plots  
plot(simple\_model)



# load the package  
library(leaps)  
  
# Fit all possible models  
all\_subsets <- regsubsets(SO2 ~ temp + manu + popul + wind + precip + predays, 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(backward\_mse, 3)))  
  
# getting the average MSE for 10 folds and 5 repeats  
bwd\_avgmse <- repeated\_cv\_error(formula(backward\_model), data = num\_pollution, 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)

