# An Investigation into the Association between Pollution

# and Human Ecology and Climate

Saf Flatters 21827361

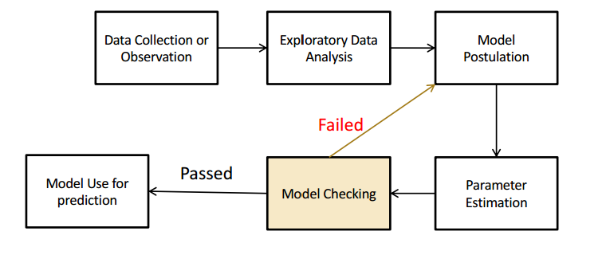
Bachelor of Science - Data Science (minors in AI & GIS)

## Introduction

Sulfur Dioxide is a key indicator of air pollution and its increasing levels can be linked to human activities and climate change. The Pollution dataset interested me particularly due to the numerical variables representing both human activity and weather patterns. In this report I analysed the data using Simple Linear Regression and Multiple Linear Regression.

My analysis (see [Appendix](#appendix) for all code and output) began with an exploration of the data provided, followed by fitting two SLR models using the ‘manu’, and ‘predays’ variables. I ensured the validity of these models by checking the residuals, assessing individual observation influence and transforming variables when required. I then used these models to create confidence and prediction intervals.

I extended my analysis to fit two MLR Models using variable selection algorithms such as ‘all subsets’ and ‘stepwise - backward selection’, ANOVA partial F-test and again checked the residuals, assessed individual observation influence and transformed variables for a better fit.

I compared all valid models (from SLR and MLR) and determined the best model. This is included that the number of manufacturing plants employing 20 or more workers, the average annual temperature in Fahrenheit, the average number of precipitation days per year, the average annual wind speed in miles per hour and if the location of that city is in the Eastern region of USA explained some of the variability of Average annual SO2 concentration in air (µg/m³)) between cities (at the time in which the data was taken from).

To ensure all necessary analysis was conducted, in a logical order, I followed the flow chart provided.

Figure 1: Flow Chart

For visualisation, I converted all my plots to using the R package ‘ggplot2’ for better readability and flexibility (Wickham et al., 2024) and assigned a colour to each explanatory variable to make it easier for my analysis and presentation to the reader. Due to the size of this report, I have put links within the text for the reader to be able to see the R output and plots mentioned (the plots included in this report are just a small subset of the ones mentioned). These variables and their assigned colours are:

• manu : number of manufacturing enterprises employing 20 or more workers - ‘**darkorange’**

• popul: population size(1970 census); in thousands - ‘**darkolivegreen’**

• precip: average annual precipitation in inches - ‘**darkslategray3’**

• predays: average number of days with precipitation per year - ‘**deeppink1’**

• temp: average annual temperature in Fahrenheit - ‘**tomato’**

• wind: average annual wind speed in miles per hour - ‘**darkorchid’**

• Region : the region in the USA - ‘**darkkhaki’**

Please note from this point forward I will refer to each variable by it’s variable name in the data to maintain conciseness. Also please note, due to the size of this report - I could not include all plots referred to, however I have included links to sections of the Appendix with important outputs throughout.

## Limitations

The analysis conducted had some significant limitations. This includes I, the analyst, not having any contextual knowledge of how the observations gathered and have had to assume the descriptions of each variable are exactly how they read and that they are independent. Another limitation is the number of observations (41). This limited the chances of detecting significant relationships, the ability to generalise and reduced chances of making meaningful conclusions. It also forced me to choose not to remove any more high leveraged outliers in the second iteration of model checks as I didn’t want to reduce the sample size any lower than I had to do in the first iteration (See [A3.4 Model Postulation - Post-Transformation & Outlier Removals](#X5f81e617ea0cf0646ee7685803f188df93f6dce)). Due to inherent issues with Linearity, Homoscedacity, Normality, Multi-collinerity and high leveraged outliers, variables underwent significant transformations to ensure validity. A limitation linked to this one would include the ability to go too far beyond the scope of this unit and use other regression techniques. However I did display an extension to this unit by using GLM Logistic Regression to analyze the association between cities in the Eastern region and SO2 readings**.**

## A2.4.1 Plotted Summary Statistics on Numerical VariablesExploration of the data

Exploration of the data started with creating and transforming the dataframe (see [A2. Exploratory Analysis](#a2.-exploratory-analysis)). I made the response variable as the first column and then put the rest of the numerical variables in alphabetical order so they would match the order they are plotted in. I also checked the structure of each column ([A2.1 Structure of Data](#a2.1-structure-of-data)), compared the first few items with the variable descriptions and determined there are 41 observations. I also looked at the summary statistics ([A2.4 Summary Statistics](#a2.4-summary-statistics)) and side-by-side boxplot output ([A2.4.1 Plotted Summary Statistics on Numerical Variables](#Xb92003f3b3a69a5834bd8d679734c90e376a1e8)) (I had to standardise the variables to do this) to see if there was any visual indication on normality, skewness and outliers (Wickham et al., 2024).

Figure 2: A2.4.1 Plotted Summary Statistics on Numerical Variables

My preliminary conclusions from visual only assessment of the summary and boxplots:

**manu** - slightly right skewed (median < mean and with a longer right tail), significant outliers with one significant outlier being more than 5 standard deviations away, the other significant outlier just above 2 standard deviations. Two less significant outliers. Quartiles are even.

**popul** - slightly right skewed (median < mean and with a longer right tail), significant outliers with one significant outlier being more than 5 standard deviations away, the other significant outlier just above 2 standard deviations. One less significant outlier. Quartiles are even.

**precip** - slightly left skewed (median > mean and with a longer left tail). Two outliers 2 standard deviations away. Negative Quartile is larger than Positive Quartile.

**predays** - not skewed at all (median mean). One significant outlier almost 3 standard deviations away and 2 non-significant outliers

**temp** - slightly right skewed (median < mean and with a longer right tail), one outlier at 2 standard deviations away. Quartiles are even.

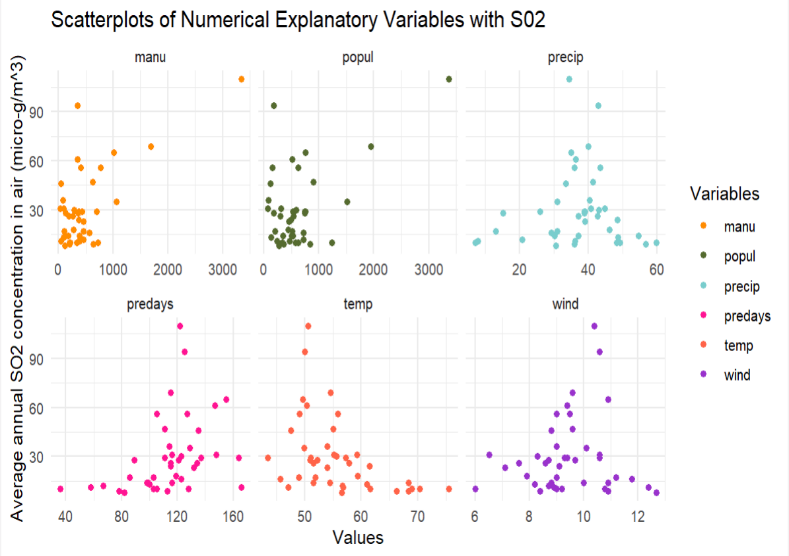
**wind** - very slightly right skewed (median < mean and with a longer right tail), no outliers. Positive Quartile is larger than Negative Quartile.

I also conducted a Shapiro-Wilk test to check normality ([A2.4.2 Normality check by Shapiro-Wilk Test](#X2d0239f226cbcf5f50be62df7557c05aaae7389)) in which the Null Hypothesis is that the data is Normally distributed. With alpha set at 0.05, the data was only normally distributed in variables ‘Predays’ and ‘Wind’. The rest were not. This confirms my visual analysis for normality of the boxplot output.

See [A2.5 Scatterplots of Numerical Variables](#a2.5-scatterplots-of-numerical-variables) for side-by-side scatterplots comparing each numerical explanatory variable (on the x axis) with the average annual SO2 (per cubic metre). Each dot is a different city in the USA. These plots may appeared clustered down one end due to extreme outliers, this will be further assessed in [A3.1.2 Model Checking](#a3.1.2-model-checking).

Visual analysis of each plot:

**manu** - Appears to be weakly positive linear as some cities with higher numbers of larger manufacturing plants reported higher average annual S02 but with a wide range of S02 values for the number of plants.

**popul** - Appears to be weakly positive linear as some cities with higher numbers of population reported higher average annual S02 but with a wide range of S02 values for the number of people. Required

**precip** - Does not appear to be linear. It’s possibly curved with no direction. Strength can not be determined either. May require transformation.

**predays** - Appears to be weakly positive and possibly linear as some cities with higher numbers of precipitation days reported higher average annual S02 but with a wide range of S02 values. This variable looks correlated with precip but shows to be slightly more linear than precip’s extreme curve.

**temp** - Appears to be weakly negatively linear or curved as some cities with higher annual temperative reported lower average annual S02.

**wind** - Does not appear to be linear. It’s possibly curved with no direction. Strength can not be determined either. May require transformation.

Figure 3: A2.5 Scatterplots of Numerical Variables

In [A2.4.4 Explore Response Variable](#a2.4.4-explore-response-variable) I looked at the normality of the response variable with a histogram and boxplot. It was significant right skewness and contained several notable outliers. I noted that the response variable may also require transformation if this could not be rectified by the explanatory transformations.

To prepare for Multiple Linear Regression I looked at a chart.Correlation plot ([A2.7 Further Exploration for MLR specifically](#Xd672ce9024652d4d8b787e561d268a39c4a559d)). Besides collinearity and correlation which I analyse in a further section of this report, I noted that not all variables appeared linear (particularly ‘predays’, ‘temp’ and ‘wind’) and this alludes to either transforming some of the variables or looking at polynomial regression as options to find the best model later.

There is only one categorical variable called ‘region’. Within this variable there are four categories East, North, South and West. I created a bar chart on the frequency counts of each category ([A2.4.3 Frequency Counts on Categorical Variable](#X4bb529c8765d14ee2034c5d201c9554c2fd986e)), made a side-by-side boxplot against the response variable and scatterplots of SO2 against each explanatory variable (with four colours to show each region category) (see [A2.6.1 Plot Categorical Variable](#a2.6.1-plot-categorical-variable)). Due to the small sample size of this data, there is limited inference that could be made about each category and it’s relationship with SO2. I did note that it was quite clear in all scatterplots that the region “East” was in the upper levels of SO2 in each plot.

In this process, I created dummy variables to separate region to prepare for linear regression ([A2.6.3 Transformation of Categorical Variables](#X9d5b4c5a074c072ca6aeb2f9ea9bbe914971620)).

## Correlation and Collinearity

To assess correlation between each explanatory variable and SO2 I performed correlation tests ([A2.5 Correlation of Numerical Variables](#a2.5-correlation-of-numerical-variables)).

The assumptions of Pearson’s Correlation Coefficient are too strict for assessing the degree of linear association for most of the numerical variables without any transformation. Pearson’s Coefficient assumes the data (as it currently is) is on equal ratio scales, that there is a linear relationship between variables and each variable has a normal distribution. After Exploration of the data, I found the only variable that fitted in with these assumptions is ‘predays’ (see [A2.5.1 Pearson Correlation Coefficient](#a2.5.1-pearson-correlation-coefficient)). To use Peason’s, I standardised the response variable ‘SO2’ so that it’s on the same ratio scale as ‘predays’. The null hypothesis is that ‘predays’ and ‘S02’ are not correlated and the alternative hypothesis being they are positively correlated. Using the test statistic, the Pearson Correlation Coefficient was 0.3696 with a p-value of 0.008. As p < (0.05), this has indicated a weak positive correlation between ‘SO2’ and ‘predays’. This also confirmed my visual analysis of the scatterplot.

For the other variables, I used Spearmans Correlation Coefficient ([A2.5.2 Spearmans Correlation Coefficient](#a2.5.2-spearmans-correlation-coefficient)) to measure the degree of monotone association because it does not assume a linear relationship or that its is normal distribution due to it’s use of ranks. There are however ties in some of this data so Spearmans have used an approximation to compute those p-values. This did not change the validity of this test. The null hypothesis for each test is that the explanatory variable and ‘S02’ are not correlated, and the alternative hypothesis is the explanatory variable and ‘S02’ are either positively or negatively correlated (based on whether the Correlation Coefficient is positive or negative). The results of a correlation with SO2 are as follows:

‘manu’ is 0.2640 and indicates a weak positive correlation, ‘popul’ is 0.08947 and indicates no correlation, ‘precip’ is -0.0026 and indicates no correlation, ‘temp’ is -0.5388 and indicates a fairly negative correlation and ‘wind’ is 0.047 and indicates a no correlation.

As regions are not an ordinal variable, we can not perform a Correlation analysis. If it was required, I would transform the SO2 variable into categorical “Low”, “Medium” and “High” and then perform a Chi-Square Test of Association (Sheather, 2009). The cut off for each ordinal variable would rely on domain knowledge and other statistical techniques outside the scope of this unit.

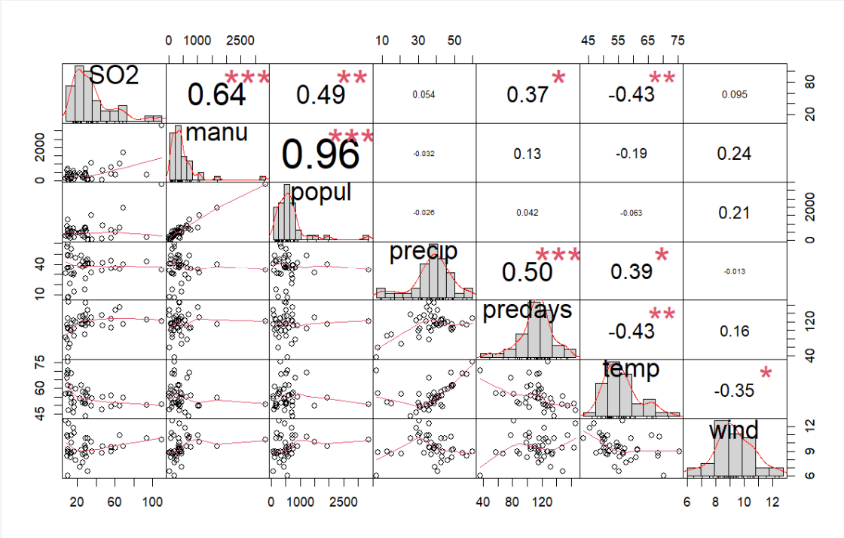
To assess collinearity for Multiple Linear Regression I created a corrplot and chart.Correlation in [A2.7 Further Exploration for MLR specifically](#Xd672ce9024652d4d8b787e561d268a39c4a559d). Both plots show ‘manu’ and ‘popul’ show a correlation of 96%. This may result in issues associated with multicollinearity such as issues with determining individual variable effects on SO2 and inflated Standard Errors (Sheather, 2009). This was later assessed by VIF (Variance Inflation Factor) during the variable selection stages of MLR (such as [A4.3.2 MLR1 Model Checking](#a4.3.2-mlr1-model-checking)). With a cut off of 5, not surprisingly both ‘manu’ and ‘popul’ scored over 14. I then re-assessed again twice, once removing ‘manu’ and again removing ‘popul’ and both times, all variables were below the cut off.

Figure 4: A2.7 Further Exploration for MLR Specifically

## Simple Linear Regression

In order to prepare for Simple Linear Regression, I wanted to reduce complexity of this analysis by choosing only variables that will be more likely to show a relationship. Due to the previous tests indicating a correlation relationship I have chosen to perform Simple Linear Regression with ‘S02’ and ‘manu’, ‘predays’, and ‘temp’. This does not discount that there may be a non-linear relationship or significant outliers that could alter the result of the variables I am using or the ones I am discarding (Ismay & Kim, 2024). I however will show transformation and outlier influence analysis on these chosen three variables once I have fitted a model. Please note that I only got SLR results for ‘manu’ and ‘predays’ as I could not reconcile the residuals to conform with SLR assumptions for ‘temp’ (even with transformations) and therefore dropped it from my SLR analysis here; [A3.2.1 Variable Transformation Assessment](#X7e5363e89d704410e32c6df6ecd3c3784a09022).

Now the data has been explored, a Simple Linear Regression model for each valid explanatory variable is to be proposed to assess the linear relationship between that explanatory variable and the response variable. The relationships ‘manu’ and ‘SO2’, ‘predays’ and ‘SO2’ and ‘temp’ and ‘SO2’ had a Line of Best Fit fitted in [A3.1.1 Parameter Estimation](#a3.1.1-parameter-estimation) using the Least Squares Method. The postulated models () created by the Line of best fit are:

**SLR1:** Average annual SO2 concentration in air (µg/m³) = 17.6106 + 0.0269 number of manufacturing plants employing 20 or more workers

**SLR2:** Average annual SO2 concentration in air (µg/m³) = -7.2270 + 0.3273 average number of days with precipitation per year

**SLR3:** Average annual SO2 concentration in air (µg/m³) = 108.5711 - 1.4081 average annual temperature in Fahrenheit

Please note that these Postulated Models are only applicable within the observed range of the observed data (Sheather, 2009). These ranges are:

* between 35 and 3344 for number of manufacturing plants employing 20 or more workers
* between 36 and 166 for average number of days with precipitation per year
* between 43.5 and 75.5 for average annual temperature in Fahrenheit.

Once the Least Squares Line was fitted and models postulated, I then tested to determine the validity of these models by checking that each model conformed to the assumptions of SLR. These assumptions are; the relationship is linear in the population, the response varies normally about the population regression line, observations are independent and the standard deviation of the responses is the same for all values of x.

During the data exploration stage, I had a preliminary suspicion that these models would not conform and transformations may be required. To check these assumptions, I conducted tests on the Residuals in [A3.1.2 Model Checking](#a3.1.2-model-checking). This was done by standardising each group of residuals and then assessing a Quantile plot and Histogram to check normality in [A3.1.2.1.1 Normality Check](#a3.1.2.1.1-normality-check), assessing a Scatterplot to check Constant Variance (homoscedacity), Linearity in [A3.1.2.1.2 Constant Variance & Linearity](#a3.1.2.1.2-constant-variance-linearity) and Type b plot (named that in R) in [A3.1.2.1.3 Residual Independence](#a3.1.2.1.3-residual-independence) to confirm there is independence between the observations (by checking it against the order of the data).

Assessing the plots I found ‘manu’, ‘predays’ and ‘temp’ all had relatively normal shaped residuals, but slightly skewed to the left. It was quite clear that manu’, ‘predays’ and ’temp did not have constant variance shown by the cluster of standardised residuals to one side of their respective scatter plots. However there was no clear indication of a lack of linearity for all three (which would be shown as a curve or some other departure. Residuals from all three variables do not appear correlated to the order of observations were (possibly) collected. This indicated that the observations were independent.

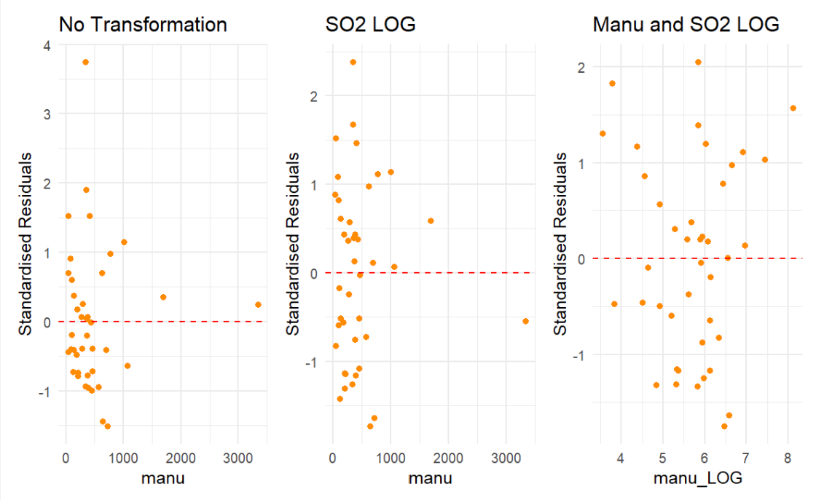
I then assessed data transformation options in [A3.2 Variable Transformation](#a3.2-variable-transformation) to ensure homoscedicity. This involved making a series of side by side scatterplots in which for each chosen variable, I assessed the effect of transforming each explanatory variable by a square-root, a cube-root or by natural log. I then assessed each of these with also transforming the response variable by these transformations as well. I also noted that ‘predays’ is a ratio of 365 days and ‘manu’ is counting data which generally suits logarithmic transformations. After assessment of these plots, I found that the SLR model for ‘manu’ conformed to SLR assumptions when ‘manu’ and ‘SO2’ had logs applied. I also found that the SLR model for ‘predays’ conformed to SLR assumptions when only ‘SO2’ had log applied.

Figure 5: One example of the many plots made in A3.2.1 Variable Transformation Assessment

Unfortunately, ‘temp’ still showed a lack of homoscedacity with all combinations of transformations (within the scope of this unit). This model was dropped from my study.

Before returning to the Model Postulation phase (since Model Checking failed), it was obvious when exploring the data that there were outliers. For an outlier to be classed as a Bad Leverage point, it must be classed as a High Leverage/Influential point by both the Hat Matrix and Cook’s Distance to be a candidate for removal (Sheather, 2009). I created plots of each of these calculations for ‘manu’ and ‘predays’. In [A3.3.1 Calculate Hat Matrix Element and Cooks Distance](#X3d131bac560f16ec739cb75c6602bc00362e365), the lime green points were assessed as ‘High Leverage’. Candidates for removal were for ‘manu’: 1, 6, 7 and for ‘predays’: 5. I assessed each leverage point one-by-one to determine if it should be removed in [A3.3.2 Assess Leverage](#a3.3.2-assess-leverage). This is done by creating a linear model of the transformed data, then removing the high leverage point and then creating another linear model. Then plotting both linear models on the same plot as well as performing an ANOVA F test and comparing the Mean Sum of Squares. A lower MSE means lower error, which in turn means a higher F-test Statistic which in turn means a lower p-value. Due to there being multiple removals, I also simulated a complete removal to ensure a model would be better without all of them. Please note: I will be performing a final ANOVA F-test in Section 3.5.1 (once all chosen candidates have been removed). All outliers identified in this first Model Checking iteration appeared to be bad leverage points and were removed in [A3.3.4 Outlier Removals](#a3.3.4-outlier-removals).

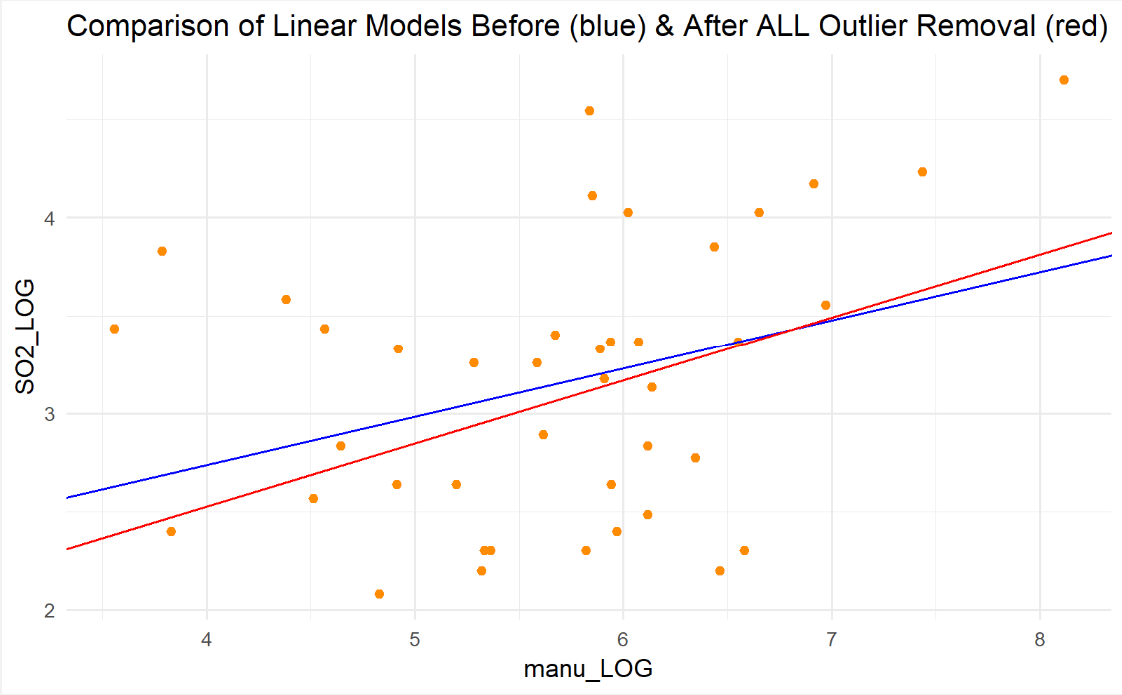
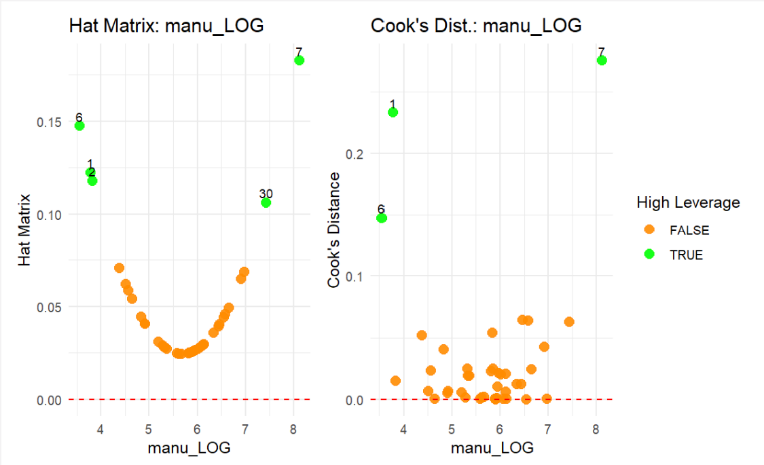


Figure 6: A3.3.1 Hat Matrix Element and Cook’s Distance

Figure 7: A3.3.2 Assess Leverage

Now the data has been transformed and outliers removed, new models were postulated for ‘manu’ and ‘predays’ in [A3.4 Model Postulation - Post-Transformation & Outlier Removals](#X5f81e617ea0cf0646ee7685803f188df93f6dce).

**SLR1:** log(Average annual SO2 concentration in air (µg/m³)) =1.2408 + 0.3219 log(number of manufacturing plants employing 20 or more workers)

**SLR2:** log(Average annual SO2 concentration in air (µg/m³)) = 1.408 + 0.0157 average number of days with precipitation per year

During the second iteration of model checking (to check these modified models conform with the assumptions of SLR) - residuals and leverage were re-assessed and leverage in [A3.4.2 Model Checking](#a3.4.2-model-checking). A new outlier not picked up in the first instance was identified in the ‘manu’ model. Due to the aforementioned limitations of this study, with such a small data set and lack of contextual knowledge, I decided not to remove this outlier.

## Simple Linear Regression Results

An ANOVA F Test was conducted on these models in [A3.5.1 ANOVA F Test](#a3.5.1-anova-f-test). This test tests whether there is a linear association between the explanatory variable and response variable (Thomas, 2023). The Null hypothesis is that there would be zero association. For the final ‘manu’ SLR model, there is some evidence that there is a non-zero linear relationship between the natural log of ‘manu’ and the natural log of ‘SO2’ (p-value = 0.019). For the final ‘predays’ SLR model, there is strong evidence that there is a non-zero linear relationship between ‘predays’ and the natural log of ‘SO2’ (p-value = 0.0001).

measures the proportion of variance in the response variable which can be explained by explanatory variables (assuming the model is linear and no bad outliers) (Sheather, 2024). In [A3.5.2 Coefficient of Determination](#a3.5.2-coefficient-of-determination), for the ‘manu’ SLR model, only 14.32% of the variance of ‘SO2’ (both transformed to natural logs) can be explained by ‘manu’. For the ‘predays’ SLR model, 32.46% of the variance of ‘SO2’ (transformed to natural log) can be explained by ‘predays’.

The final models, back-transformed are:

**SLR1:** Average annual SO2 concentration in air (µg/m³) = number of manufacturing plants employing 20 or more workers

**SLR2:** Average annual SO2 concentration in air (µg/m³) = where is the average number of days with precipitation per year

In [A3.5.3.1 Confidence and Prediction Intervals in Logarithmic Space](#Xf2d1dfdf23491a3dbd6b6a219598bc4863e851b), the two plots shows the confidence and prediction intervals for the “observed x” plotted with the linear regression model for each variable in the logarithmic space. In [A3.5.3.2 Confidence and Prediction Intervals in Exponentiated Space](#Xb3a948fb9508205c7061b563cd780a533f9a27a), the two plots shows the confidence and prediction intervals for the “observed x” plotted with the linear regression model for each variable after being back transformed. See below confidence & prediction plots for the ‘predays’ in Logarithmic space and not.

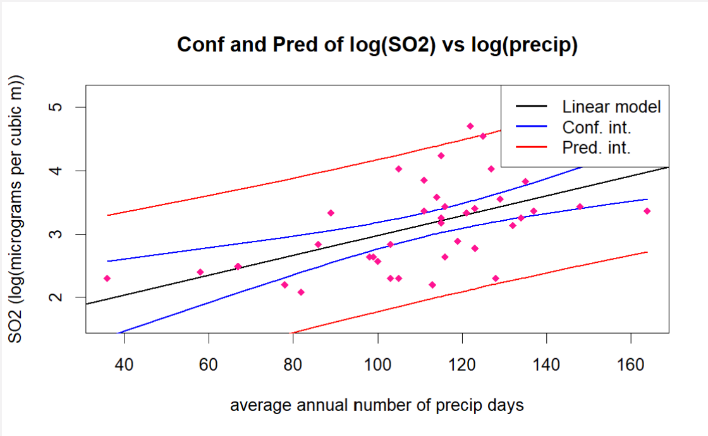
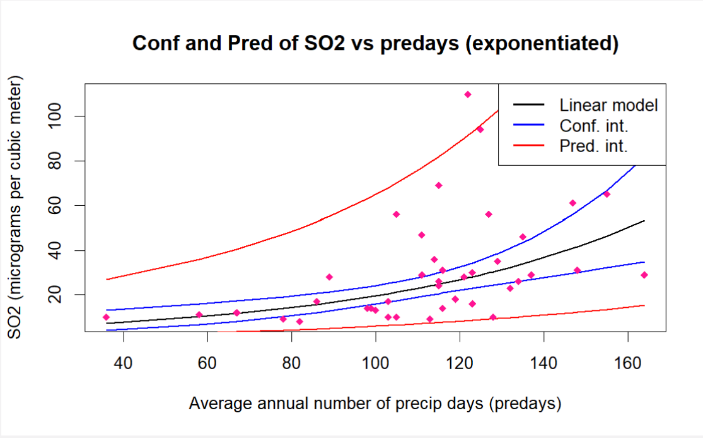


Figure 8: A3.5.3.1 Conf & Pred Intervals in Logarithmic Space

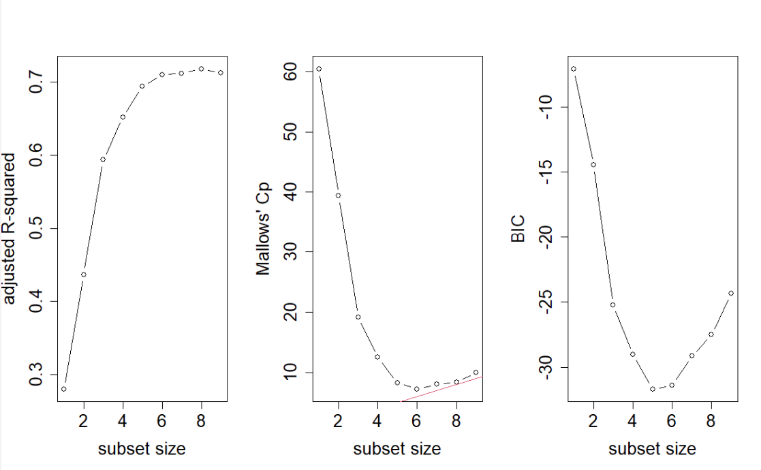
Figure 9: A3.5.3.2 Conf & Pred Intervals in Exponential Space

## Multiple Linear Regression

During the Multiple Linear Regression phase of this project, I created two final MLR models and compared them.

To begin MLR, I fitted a model using all the data provided (including the one-hot encoded categorical variable) to provide a baseline in [A4.1 Fit Model using all Explanatory Variables](#Xa6d2aad4bd754dd87eb793b8d163160a2b8c5ab). This very likely would not meet the linearity, homoscedacity, normality and multi-collinearity assumptions of MLR. It was noted in the summary that the adjusted was 69% (without meeting assumptions) and only 1 variable (‘manu’) has a p-value of under 0.05 (even the intercept was deemed not significant!). In [A2.7 Further Exploration for MLR specifically](#Xd672ce9024652d4d8b787e561d268a39c4a559d) I performed a VIF test with all the variables that was used in this first baseline fitting for MLR. Only two variables showed significance amongst others but it’s important to note these two variables ‘manu’ and ‘popul’ are highly correlated (96%) and failed the VIF test when included together. SO2 presented itself as a potential for natural logarithmic transformation during the exploratory process (and confirmed during SLR). I transformed it as a preliminary measure to compare baselines. This increased the number of variables that are significant (‘manu’, ‘temp’ and ‘wind’) and adjusted = 0.71% (still without meeting assumptions).

To compare variable selection techniques in [A4.2 Variable Selection](#a4.2-variable-selection), I first used Stepwise Backwards Direction in [A4.2.1 Stepwise](#a4.2.1-stepwise) and then as a comparison I used [A4.2.2 All Subsets](#a4.2.2-all-subsets) (Brute Force) which allowed me to interpret the data more manually during the variable selection process. Although once using All Subsets (with subset size plots of Adjusted , Mallows’ and BIC recommending subset sizes of 5 or 6) the p-value was slightly worse, I found that all 5 variables included were ‘significant’. Stepwise chose ‘manu’ and ‘popul’ which with a high VIF would have skewed the and standard errors and many of the variables individually were not ‘significant’. While stepwise selection initially appear to offer better overall fit, the All Subsets selection’s advantage of lower individual p-values and avoidance of highly correlated variables makes it a better choice for building a reliable and interpretable model.

The first MLR model postulated in [A4.3.1 MLR1 Model Postulation - Pre-Transformation](#X8539e0c7c592769243453213c29d2aa03589d54) selected: ‘manu’, ‘temp’, ‘predays’, ‘wind’ and ‘region\_East’ with an adjusted of 0.6934 and a p-value of 0.0000 (3.94e-09). This is that the:

Average annual SO2 concentration in air (µg/m³) = 5.6298 + 0.00055 number of manufacturing plants employing 20 or more workers + (-0.0411) average annual temperature in Fahrenheit + 0.0063 average number of precipitation days per year + (-0.1366) average annual wind speed in miles per hour + if the region in the USA is “East”: 0.6210

The strongest effect per unit change is if the region in the USA is “East” (steepest slope). The strongest effect overall in number of manufacturing plants (highest t-score) and that is also the most statistically significant (lowest p-value).

Figure 10: A4.2.2 All Subsets

I conducted checks on the standardised residuals in [A4.3.3 MLR1 Checking linear regression assumptions with residuals](#X879477378c60380ae8b362288604ef1e4d7e7cc) which included checking normality with a Quartile plot and histogram, checking linearity and homoscedacity with a scatterplots (vs Fitted Values), assessing Leverage with Cook’s Distance and Hat Matrix.

Besides making SO2 logarithmic, I originally made ‘manu’ logarithmic too as the ‘manu’ histogram was skewed and it may have an affect on the deviation of the residuals on the qqplot. This rectified issues with normality but I found the trade-off of a lower and p-value too severe. After doing some research, I found that converting ‘manu’ to a polynomial may be a better option if it satisfied the assumptions which it did (just not as normal as a logarithm would make it) (Sheather, 2009). By transforming ‘manu’ with a cube-root in [A4.3.4 MLR1 Transformation](#a4.3.4-mlr1-transformation), I decreased and increased p-value. This is a trade-off to ensure that the assumptions of the model are met (which I checked in [A4.3.5 MLR1 Checking assumptions post\_transformation](#X67d81d17671aa9e5959eb33e94224a5f593dc45)).

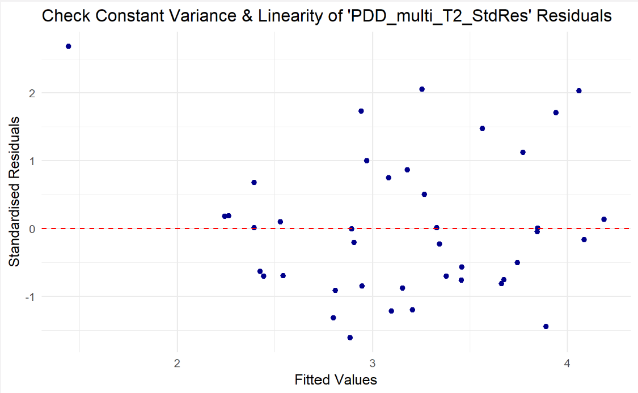
I postulated a second MLR model in which I added converted wind as a polynomial (using both the original wind, the square root of wind and the cube root of wind). Using the ANOVA Partial F test [A4.4.2 Compare MLR2 Model to MLR1 Model - Partial F-test](#X51b5972001228d933a0789b8dacc49bfe7e2c32) to determine if the increased model would make a better fit - with a p-value of 0.037, I rejected the Null Hypothesis indicating that the additional predictors contributed to the model. The linear model then went through the same assessments as above here: [A4.4.3 MLR2 Checking linear regression assumptions](#X9247fdff7fbb7a86ab652cf56d07423e9a7e4a9) and resulted in removing a High Leverage point that increased normality and homoscedacity here [A4.4.4 Outlier Removal and reassess linear assumptions with residuals](#Xe9ba4776f8d76be22eb288621274ce14169fef9).

Figure 11: A4.4.3 Checking homoscedacity

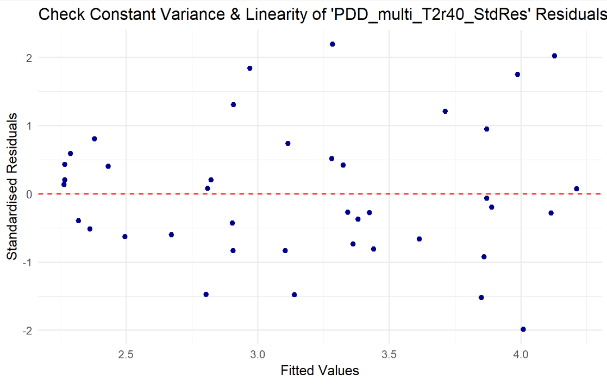


Figure 12: A4.4.4 Outlier Removal and Reassess homoscedacity

## Multiple Linear Regression Results

**Final MLR Model 1**:

Log(Average annual SO2 concentration in air (µg/m³)) = 5.294 + (number of manufacturing plants employing 20 or more workers) -0.0433 (average annual temperature in Fahrenheit) + 0.0059 (average number of precipitation days per year) -0.1461 (average annual wind speed in miles per hour) + if in the “East” region of USA is “East”: +0.6179

The summary and ANOVA test in [A4.3.6 MLR1 Final Model 1 & ANOVA](#a4.3.6-mlr1-final-model-1-anova) shows that approximately 64.36% of the variability in SO2 levels are explained, as indicated by the adjusted , with a p-value less than 0.01, suggesting a statistically significant relationship. The ANOVA test determined this model’s overall significance where where at least 1 does not equal 0. Each variable had a p-value of less that 0.05 concluding that each variable contributes to explaining the variability of SO2.

**Final MLR Model 2:**

Log(Average annual SO2 concentration in air (µg/m³)) = 3.942 + (number of manufacturing plants employing 20 or more workers) - 0.045 (average annual temperature in Fahrenheit) + 0.0089 (average number of precipitation days per year) - 1.2048 (average annual wind speed in miles per hour) - (average annual wind speed in miles per hour) - (average annual wind speed in miles per hour) + if in the “East” region of USA is “East”: +0.5397

In [A4.4.5 MLR2 Final Model 2 & ANOVA](#a4.4.5-mlr2-final-model-2-anova), this model explains approximately 74.13% of the variability in SO2 levels, as indicated by the adjusted , with a p-value less than 0.01, suggesting a statistically significant relationship. I conducted an ANOVA test to determine this model’s overall significance where <- at least 1 does not equal 0. Each variable had a p-value of less that 0.05 concluding that each variable contributes to explaining the variability of SO2.

**The MLR Model 2 is the better model.**

## Discussion / Conclusion

In conclusion, my analysis utilising both Simple Linear Regression (SLR) and Multiple Linear Regression (MLR) provides valuable insights into the factors affecting air pollution (sulfur dioxide concentration). The transition from SLR to MLR enabled the incorporation of multiple variables, which significantly improved the model’s ability to predict average annual SO2 concentrations in various cities. My best model, which achieved an of 74%, shows that key factors such as the number of manufacturing plants, average annual temperature, precipitation days, and wind speed all play critical roles in determining SO2 pollution levels. Additionally, being in the eastern region of the U.S. was also associated with a significant increase in SO2 concentrations. As mentioned in the Limitations sections, this model may not fully capture the complexity of the factors influencing sulfur dioxide (SO2) concentrations due to several constraints inherent in the analysis. The assumptions made regarding the independence and description of variables could lead to potential biases or misinterpretations, especially without contextual knowledge of the data collection process. The relatively small sample size of 41 observations and restricting the analysis to the scope taught in first year, limited the statistical power of the analysis, making it difficult to detect significant relationships and generalize the findings to a broader context.

However using the model found in my Mutliple Linear Regression analysis, we can predict levels and develop targeted strategies to an extent to mitigate pollution and climate change in the USA. To continue this analysis further (and if I wasn’t constrained by report size and time), I would look into measuring the predictive abilities of my MLR model (such as using the *PRESS* statistic in Lecture 12).

## Extension of Study

As an extension to the above study, I looked into other regression techniques to make statistical inferences. As some of the data is non-linear in nature, I first attempted a non-linear regression but found it to be a lot of duplication of what I had achieved above with data transformation. As I am also studying Geospatial science as well as Data Science, I wanted to perform more analysis on the region category. I aimed to see if there was an association between living in the Eastern region of USA and average annual SO2. I made a logistic regression model ([A5 Extension](#a5-extension)) using Generalised Linear Model function and made the dummy variable I made for ‘region\_East’ the response variable (where 1 is East and 0 is not East) and SO2 the explanatory variable (Hosmer & Lemeshow, 2000).

The postulated model is:

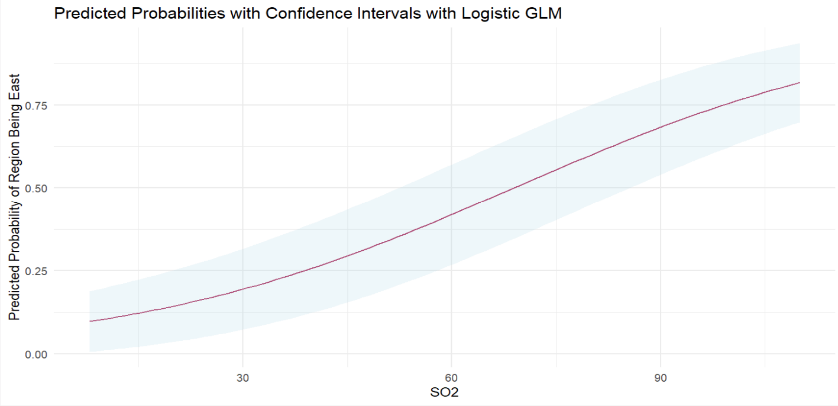
The Intercept having a pvalue < 0.01 and SO2 having a pvalue < 0.05 suggests that SO2 is a significant predictor of whether a pollution reading was taken in the East. The residual deviance being lower than the Null deviance also suggests that SO2 improves the model fit (Bruin, 2006). I then calculated confidence intervals based on the profiled log-likelihood function, plotted predicted probabilities and checked high leverage outliers (there was one but due to the size of this project, this is where I stopped for now). I found through my research (Homer & Lemeshow, 2000) however, that logit models generally require large sample sizes to be accurate.

Figure 13: A5 Extension Logistic GLM Predicted Probabilities. and Confidence Intervals

## References

Bruin, J. (2006) *Logit Regression.* UCLA: Statistical Consulting Group. <https://stats.oarc.ucla.edu/r/dae/logit-regression/>

Hosmer, D. & Lemeshow, S. (2000). Applied Logistic Regression (Second Edition). New York: John Wiley & Sons, Inc.

Ismay, C. & Kim, A. Y. (2024). Statistical Inference via Data Science: A Modern Dive into R and the Tidyverse (Second Edition). CRC Press. https://moderndive.com/v2/index.html

Kosourova, E. (2022). *How to write functions if R*. DataQuest. https://www.dataquest.io/blog/write-functions-in-r/

Sheather, S. (2009). *A Modern Approach to Regression with R.* Springer. DOI: 10.1007/978-0-387-09608-7

Curtin University. (2024) *STAT1006 Lecture Slide packs*

Curtin University. (2018) *STAT1006 Lecture Slide packs*

Thomas, S. (2023). *What is ANOVA Analysis?* Outlier. <https://articles.outlier.org/anova-analysis>

Wickham, H., Chang, C., Henry, L., Pederson, T. L., Takahashi, K., Wilke, C., Woo, K., Yutani, H., Dunnington, D., & Brand, T. (2024). *Introduction to ggplot2.* ggplot2. <https://ggplot2.tidyverse.org/reference/ggplot.html>

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, Dewey Dunnington, and Teun van den Brand. (2024) *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. Ggplot2. <https://ggplot2.tidyverse.org>.

## Appendix

#install.packages("ggplot2")  
# install and load packages  
library(ggplot2)  
library(tidyr)  
library(dplyr)  
library(tibble)  
library(patchwork) #plot side by side  
library(PerformanceAnalytics)   
library(car)  
library(corrplot)  
library(leaps)  
library(tinytex)

## Appendix Functions

#####CORRELATIONS  
  
#No need to use standardised variables  
#Function to perform Spearman Correlation Hypothesis test (called by spearman\_corrFunction)  
spearman\_testFunction <- function(xvar, alternative) {  
 #suppress warnings for ties  
 spearmantest\_result <- suppressWarnings(  
 cor.test(PollutionData[[xvar]], PollutionData$SO2, method = "spearman", alternative = alternative)  
 )  
 #extract to print p value  
 p\_value <- spearmantest\_result$p.value  
 #test\_statistic <- spearmantest\_result$statistic  
 #print the result of the hypothesis test  
 cat("Spearman Hypothesis test of", xvar, "and SO2: where H\_1 is", alternative, "\n",  
 "P-value:", p\_value, "\n\n")  
}  
#Function to perform Spearman Correlation Coefficient and call for hyp test function  
spearman\_corrFunction <- function(xvar) {  
 spearmancorr\_result <- cor(PollutionData[[xvar]], PollutionData$SO2, method = "spearman")  
 #print correlation number  
 cat("Spearman Correlation of", xvar, "and SO2: ", spearmancorr\_result, "\n")  
 #call hypothesis test depending if alternative should be greater or less  
 if (spearmancorr\_result > 0) {  
 spearman\_testFunction(xvar, "greater")  
 } else {   
 spearman\_testFunction(xvar, "less")  
 }  
}  
  
#### SLR LINE OF BEST FIT  
  
#functions to fit line, print summary and plot  
fitline\_function <- function(xvar, data) {  
 # Dynamic formula for lm  
 formula <- as.formula(paste("SO2 ~", xvar))  
 # Fit the linear model  
 lm\_model <- lm(formula, data = data)  
 return(lm\_model)  
}   
lmsummary\_function <- function(xvar, lm\_model) {  
 return(summary(lm\_model))  
}  
lmplot\_function <- function(xvar, colour, data) {  
   
 plot\_lm <- plot(ggplot(data, aes\_string(x = xvar, y = "SO2")) +  
 geom\_point(color = colour) +  
 geom\_smooth(method = "lm", color = "blue", se = FALSE) +  
 theme\_minimal()   
 )  
 return(plot\_lm)  
}  
  
  
### CHECK RESIDUALS   
  
# Function to create a Q-Q plot using ggplot  
ggplot\_qq <- function(residuals, title, color) {  
 ggplot(data.frame(sample = residuals), aes(sample = sample)) +  
 stat\_qq(color = color) +  
 stat\_qq\_line(color = "black", linetype = "dashed") +  
 labs(title = title) +  
 theme\_minimal()  
}  
# Function to create a ggplot histogram with a normal curve   
ggplot\_hist\_with\_normal <- function(residuals, title, xlab, fill\_color) {  
 # Create a data frame for ggplot to use  
 residuals\_df <- data.frame(residuals = residuals)  
 ggplot(residuals\_df, aes(x = residuals)) +  
 geom\_histogram(aes(y = ..density..), bins = 20, fill = fill\_color, color = "black", alpha = 0.7) +  
 stat\_function(fun = dnorm,   
 args = list(mean = mean(residuals, na.rm = TRUE),   
 sd = sd(residuals, na.rm = TRUE)),   
 color = "red", size = 1) +  
 labs(title = title, x = xlab, y = "Density") +  
 theme\_minimal()  
}  
  
### CHECKING TRANSFORMATIONS  
  
#FIT LM MODEL  
make\_linear\_model\_function <- function(data, xvar, yvar) {  
 formula <- as.formula(paste(yvar,"~", xvar))  
 linear\_model <- lm(formula, data = data)   
 return(linear\_model)  
}  
#Standardise Residuals  
standard\_residuals\_function <- function(data, xvar, yvar) {  
 lmodel <- make\_linear\_model\_function(data, xvar, yvar)  
 std.residuals <- rstandard(lmodel)  
 return(std.residuals)  
}  
#Scatterplot residuals to check homdcesdacity and linearity  
residual\_plot\_function <- function(data, xvar, yvar, title, colour) {   
 yy <- standard\_residuals\_function(data, xvar, yvar)  
 lmplot <- ggplot(data, aes\_string(x = xvar, y = "yy")) +  
 geom\_point(color = colour) +   
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +   
 labs(x = xvar, y = "Standardised Residuals",   
 title = title) +  
 theme\_minimal()   
}  
  
  
####LEVERAGE  
  
#Hat Values  
plot\_hat\_values <- function(data, x\_var, y\_var, false\_color = "darkorange") {   
 # Fit the linear model using the specified x variable   
 model\_formula <- as.formula(paste(y\_var, "~", x\_var))   
 model <- lm(model\_formula, data = data)   
 # Calculate the hat values   
 hat\_values <- hatvalues(model)   
 #Calculate p  
 p <- length(coef(model)) - 1  
 #Calculate n  
 n <- nrow(data)  
 # Combine hat values with the original data   
 data <- data %>%   
 mutate(HatValues = hat\_values,   
 HighLeverage = HatValues > (2\*(p + 1) / n())) %>%  
 mutate(ID = row\_number())  
 # Create the plot   
 ggplot(data, aes\_string(x = x\_var, y = "HatValues")) +   
 geom\_point(aes(color = HighLeverage),   
 alpha = 0.9,   
 size = 3,  
 show.legend = FALSE) +   
 scale\_color\_manual(values = c("FALSE" = false\_color, "TRUE" = "green"), name = "High Leverage") + # Custom colors for high leverage   
 geom\_hline(yintercept = 0, linetype = "dashed", color = "red") + # Horizontal line at y = 0   
 labs(x = x\_var, y = "Hat Matrix",   
 title = paste("Hat Matrix:", x\_var)) +   
 theme\_minimal() +   
 # Add labels for high leverage points   
 geom\_text(data = filter(data, HighLeverage),   
 aes(label = ID),   
 vjust = -0.5,   
 color = "black",   
 size = 3) # Adjust size as needed   
}  
#Cooks Distance  
plot\_cooks\_distance <- function(data, x\_var, y\_var, false\_color = "darkorange") {   
 # Fit the linear model using the specified x variable   
 model\_formula <- as.formula(paste(y\_var,"~", x\_var))   
 model <- lm(model\_formula, data = data)   
 # Calculate Cook's distance   
 cooks\_distance <- cooks.distance(model)   
 #Calculate p  
 p <- length(coef(model)) - 1  
 #Calculate n  
 n <- nrow(data)  
 # Combine Cook's distance with the original data   
 data <- data %>%   
 mutate(CooksDistance = cooks\_distance,   
 HighInfluence = CooksDistance > (2\*(p + 1) / (n() - (p -2)))) %>%  
 mutate(ID = row\_number())   
 # Create the plot   
 ggplot(data, aes\_string(x = x\_var, y = "CooksDistance")) +   
 geom\_point(aes(color = HighInfluence),   
 alpha = 0.9,   
 size = 3) +   
 scale\_color\_manual(values = c("FALSE" = false\_color, "TRUE" = "green"), name = "High Leverage") + # Custom colors for high influence   
 geom\_hline(yintercept = 0, linetype = "dashed", color = "red") + # Horizontal line at y = 0   
 labs(x = x\_var, y = "Cook's Distance",   
 title = paste("Cook's Dist.:", x\_var)) +   
 theme\_minimal() +   
 # Add labels for high influence points   
 geom\_text(data = filter(data, HighInfluence),   
 aes(label = ID),   
 vjust = -0.5,   
 color = "black",   
 size = 3) # Adjust size as needed   
}   
  
### COMPARE SLR MODELS   
  
compare\_lm\_function <- function(data, x\_var, y\_var, row\_to\_remove, colour) {  
 # Step 1: Fit the initial linear model  
 model\_formula <- as.formula(paste(y\_var, "~", x\_var))  
 initial\_model <- lm(model\_formula, data = data)  
 # Step 2: Remove the specified row  
 data\_no\_outlier <- data[-row\_to\_remove, ]  
 # Step 3: Fit the linear model after removing the row  
 model\_no\_outlier <- lm(model\_formula, data = data\_no\_outlier)  
 # Step 4: Plot both linear models  
 plot <- ggplot(data, aes\_string(x = x\_var, y = y\_var)) +  
 geom\_point(color = colour) +  
 geom\_abline(intercept = coef(initial\_model)[1],   
 slope = coef(initial\_model)[2],   
 color = "blue") +  
 geom\_abline(intercept = coef(model\_no\_outlier)[1],   
 slope = coef(model\_no\_outlier)[2],   
 color = "red") +  
 theme\_minimal() +  
 labs(title = "Comparison of Linear Models Before (blue) and After Outlier Removal (red)",  
 x = x\_var,  
 y = y\_var)  
 print(plot)  
 # Step 5: Calculate ANOVA and print results  
 anova\_initial <- anova(initial\_model)  
 anova\_no\_outlier <- anova(model\_no\_outlier)  
 mse\_initial <- anova\_initial["Residuals", "Mean Sq"]  
 mse\_no\_outlier <- anova\_no\_outlier["Residuals", "Mean Sq"]  
 cat("\n##### Compare ANOVA Tables for", x\_var, "Outlier: ID", row\_to\_remove)  
 cat("\nPre-Outlier-Removal:")  
 cat("\nMSE:", mse\_initial)  
 cat("\nPost-Outlier-Removal:")  
 cat("\nMSE:", mse\_no\_outlier)  
 # Step 6: Determine which model fits better  
 if (mse\_no\_outlier < mse\_initial) {  
 cat("\n\nModel after Outlier Removal fits better (due to lower MSE = higher F = lower pvalue).\n")  
 } else {  
 cat("\n\nModel before Outlier Removal fits better (due to lower MSE = higher F = lower pvalue).\n")  
 }  
}

## A1. Introduction

### A1.1 Setting up

PollutionDf <- read.csv("Pollution.csv")  
  
#Make Dataframe show numerical variables in alphabetical order so plots all match  
alphorder <- c("SO2", "manu", "popul", "precip", "predays", "temp", "wind", "region")  
PollutionData <- PollutionDf[alphorder]  
  
PollutionData <- PollutionData %>%   
 mutate(ID = row\_number())  
  
# View(PollutionData)  
head(PollutionData)

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

n <- nrow(PollutionData) #number of observations

## A2. Exploratory Analysis

### A2.1 Structure of Data

str(PollutionData)

'data.frame': 41 obs. of 9 variables:  
 $ SO2 : int 46 11 24 47 11 31 110 23 65 26 ...  
 $ manu : int 44 46 368 625 391 35 3344 462 1007 266 ...  
 $ popul : int 116 244 497 905 463 71 3369 453 751 540 ...  
 $ precip : num 33.36 7.77 48.34 41.31 36.11 ...  
 $ predays: int 135 58 115 111 166 148 122 132 155 134 ...  
 $ temp : num 47.6 56.8 61.5 55 47.1 55.2 50.6 54 49.7 51.5 ...  
 $ wind : num 8.8 8.9 9.1 9.6 12.4 6.5 10.4 7.1 10.9 8.6 ...  
 $ region : chr "East" "South" "East" "East" ...  
 $ ID : int 1 2 3 4 5 6 7 8 9 10 ...

### A2.2 Variables

### A2.3 Standardised Numerical Variables

#Standardise Numerical Exp Data so it can be visually checked on same plots  
PD\_std <- PollutionData %>%  
 transmute(  
 SO2 = SO2,  
 temp\_std = scale(temp)[,1],  
 manu\_std = scale(manu)[,1],  
 popul\_std = scale(popul)[,1],  
 wind\_std = scale(wind)[,1],  
 precip\_std = scale(precip)[,1],  
 predays\_std = scale(predays)[,1],  
 region = region,  
   
 )  
# View(PD\_std)

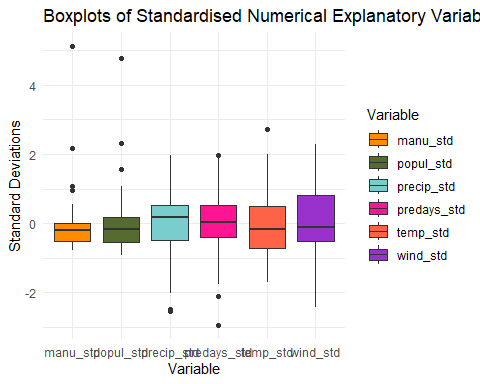
### A2.4 Summary Statistics

summary(PollutionData)

SO2 manu popul precip   
 Min. : 8.00 Min. : 35.0 Min. : 71.0 Min. : 7.05   
 1st Qu.: 13.00 1st Qu.: 181.0 1st Qu.: 299.0 1st Qu.:30.96   
 Median : 26.00 Median : 347.0 Median : 515.0 Median :38.74   
 Mean : 30.05 Mean : 463.1 Mean : 608.6 Mean :36.77   
 3rd Qu.: 35.00 3rd Qu.: 462.0 3rd Qu.: 717.0 3rd Qu.:43.11   
 Max. :110.00 Max. :3344.0 Max. :3369.0 Max. :59.80   
 predays temp wind region   
 Min. : 36.0 Min. :43.50 Min. : 6.000 Length:41   
 1st Qu.:103.0 1st Qu.:50.60 1st Qu.: 8.700 Class :character   
 Median :115.0 Median :54.60 Median : 9.300 Mode :character   
 Mean :113.9 Mean :55.76 Mean : 9.444   
 3rd Qu.:128.0 3rd Qu.:59.30 3rd Qu.:10.600   
 Max. :166.0 Max. :75.50 Max. :12.700   
 ID   
 Min. : 1   
 1st Qu.:11   
 Median :21   
 Mean :21   
 3rd Qu.:31   
 Max. :41

#### A2.4.1 Plotted Summary Statistics on Numerical Variables

# Reshape data for boxplot  
long\_data <- PD\_std %>%  
 pivot\_longer(  
 cols = !c(SO2, region), # Exclude columns SO2 and region  
 names\_to = "Variable",  
 values\_to = "Value"  
 )  
  
# Create side-by-side boxplots  
ggplot(long\_data, aes(x = Variable, y = Value, fill = Variable)) +  
 geom\_boxplot() +   
 theme\_minimal() +  
 labs(title = "Boxplots of Standardised Numerical Explanatory Variables", y = "Standard Deviations") +  
 scale\_fill\_manual(values = c('darkorange', 'darkolivegreen','darkslategray3','deeppink1','tomato', 'darkorchid'))



#### A2.4.2 Normality check by Shapiro-Wilk Test

# hypothesis testing for normality, Ho: data are Normal   
  
shapiro.test(PD\_std$manu\_std)

Shapiro-Wilk normality test  
  
data: PD\_std$manu\_std  
W = 0.60548, p-value = 2.781e-09

shapiro.test(PD\_std$popul\_std)

Shapiro-Wilk normality test  
  
data: PD\_std$popul\_std  
W = 0.68049, p-value = 3.623e-08

shapiro.test(PD\_std$precip\_std)

Shapiro-Wilk normality test  
  
data: PD\_std$precip\_std  
W = 0.94214, p-value = 0.03725

shapiro.test(PD\_std$predays\_std)

Shapiro-Wilk normality test  
  
data: PD\_std$predays\_std  
W = 0.9654, p-value = 0.2419

shapiro.test(PD\_std$temp\_std)

Shapiro-Wilk normality test  
  
data: PD\_std$temp\_std  
W = 0.93554, p-value = 0.02215

shapiro.test(PD\_std$wind\_std)

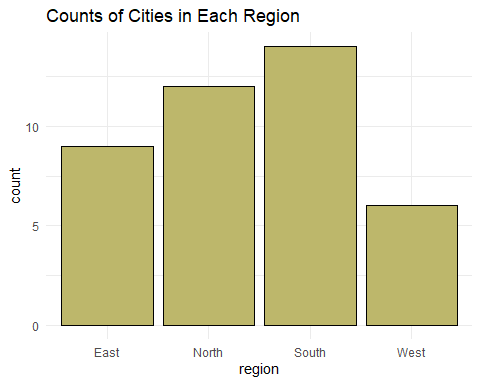
Shapiro-Wilk normality test  
  
data: PD\_std$wind\_std  
W = 0.98057, p-value = 0.6973

#### A2.4.3 Frequency Counts on Categorical Variable

table(PollutionData$region)

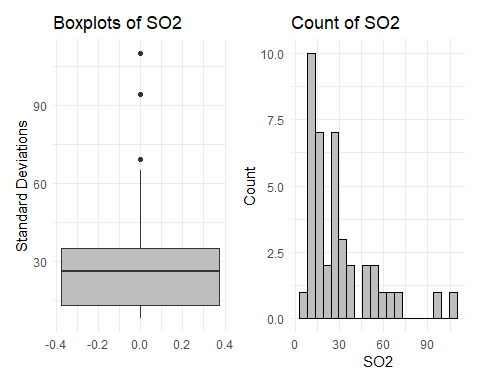
East North South West   
 9 12 14 6

ggplot(data = PollutionData, mapping = aes(x = region)) +  
 theme\_minimal() +  
 labs(title = "Counts of Cities in Each Region") +  
 geom\_bar(color = "black", fill="darkkhaki")



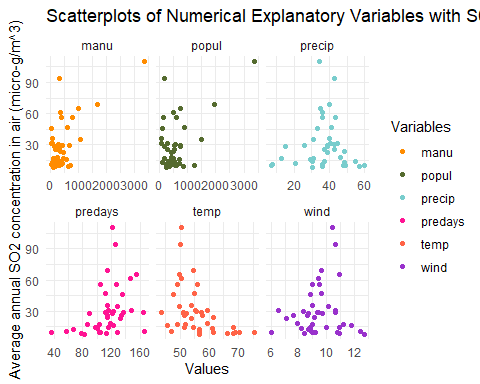
#### A2.4.4 Explore Response Variable

SO2box <- ggplot(PD\_std, aes(y = SO2)) +  
 geom\_boxplot(fill = 'grey') +   
 theme\_minimal() +  
 labs(title = "Boxplots of SO2", y = "Standard Deviations")  
  
SO2hist <- ggplot(PollutionData, aes(x = SO2)) +   
 geom\_histogram(fill = "grey", color="black", bins=20) +   
 theme\_minimal() +  
 labs(title = "Count of SO2",   
 x = "SO2",   
 y = "Count")  
  
SO2box + SO2hist



### A2.5 Scatterplots of Numerical Variables

# Reshape the data from wide to long format  
long\_data2 <- PollutionData %>%  
 pivot\_longer(cols = c(temp, manu, popul, wind, precip, predays),  
 names\_to = "Variables",  
 values\_to = "Value2")  
  
# Create scatterplots with facet\_wrap to see all variables in same plot  
ggplot(data = long\_data2, aes(x = Value2, y = SO2, color = Variables)) +  
 geom\_point() +  
 theme\_minimal() +  
 facet\_wrap(~ Variables, scales = "free\_x") +  
 labs(title = "Scatterplots of Numerical Explanatory Variables with S02", x = "Values", y = "Average annual SO2 concentration in air (micro-g/m^3) ") +   
 scale\_color\_manual(values = c('darkorange', 'darkolivegreen','darkslategray3','deeppink1','tomato', 'darkorchid'))



### A2.5 Correlation of Numerical Variables

#### A2.5.1 Pearson Correlation Coefficient

#Standardise response variable  
SO2\_std <- scale(PD\_std$SO2)  
  
#Pearson Correlation Coefficient of predays  
cor(PD\_std$predays\_std,  
 SO2\_std,  
 method= "pearson"  
 )

[,1]  
[1,] 0.3695636

#Hypothesis test for Correlation of predays  
cor.test(x=PD\_std$predays\_std, y=SO2\_std,  
 alternative="greater", # H\_A: true correlation is greater than zero  
 method= "pearson"  
 )

Pearson's product-moment correlation  
  
data: PD\_std$predays\_std and SO2\_std  
t = 2.4838, df = 39, p-value = 0.008702  
alternative hypothesis: true correlation is greater than 0  
95 percent confidence interval:  
 0.1204988 1.0000000  
sample estimates:  
 cor   
0.3695636

#### A2.5.2 Spearmans Correlation Coefficient

#Spearman Correlation between each exp variable and response variable  
spearman\_corrFunction("manu")

Spearman Correlation of manu and SO2: 0.2640507   
Spearman Hypothesis test of manu and SO2: where H\_1 is greater   
 P-value: 0.04763628

spearman\_corrFunction("popul")

Spearman Correlation of popul and SO2: 0.0894703   
Spearman Hypothesis test of popul and SO2: where H\_1 is greater   
 P-value: 0.2890062

spearman\_corrFunction("precip")

Spearman Correlation of precip and SO2: -0.002616091   
Spearman Hypothesis test of precip and SO2: where H\_1 is less   
 P-value: 0.4935242

spearman\_corrFunction("temp")

Spearman Correlation of temp and SO2: -0.5388308   
Spearman Hypothesis test of temp and SO2: where H\_1 is less   
 P-value: 0.0001392113

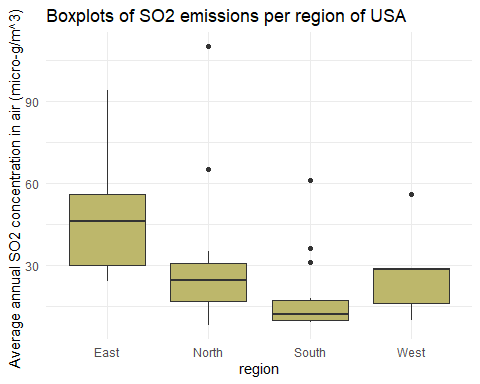
spearman\_corrFunction("wind")

Spearman Correlation of wind and SO2: 0.04730939   
Spearman Hypothesis test of wind and SO2: where H\_1 is greater   
 P-value: 0.3844843

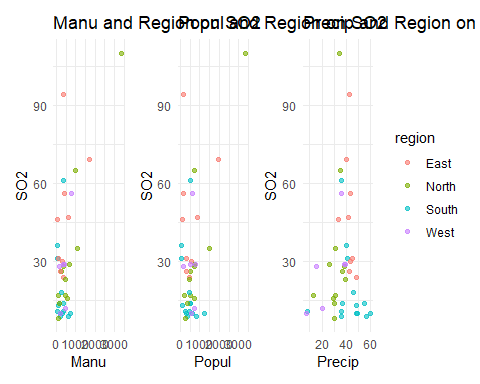
### A2.6 Explore Categorical Variable

#### A2.6.1 Plot Categorical Variable

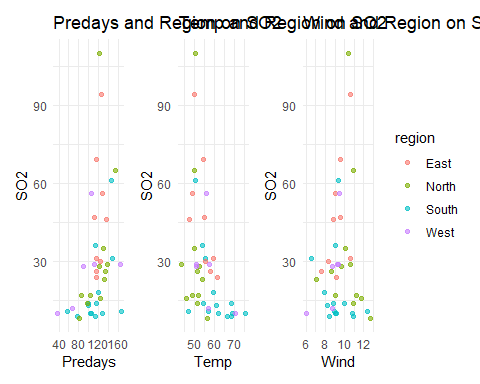
# Create side-by-side boxplots  
ggplot(PollutionData, aes(x = region, y = SO2)) +  
 geom\_boxplot(fill = "darkkhaki") +   
 theme\_minimal() +  
 labs(title = "Boxplots of SO2 emissions per region of USA", y = "Average annual SO2 concentration in air (micro-g/m^3) ")



r1 <- ggplot(PollutionData, aes(x = manu, y = SO2, color = region)) +  
 geom\_point(alpha = 0.6) +   
 labs(title = "Manu and Region on SO2",   
 x = "Manu",   
 y = "SO2") +  
 theme\_minimal() +   
 theme(legend.position = "none")  
r2 <- ggplot(PollutionData, aes(x = popul, y = SO2, color = region)) +  
 geom\_point(alpha = 0.6) +   
 labs(title = "Popul and Region on SO2",   
 x = "Popul",   
 y = "SO2") +  
 theme\_minimal()+   
 theme(legend.position = "none")  
r3 <- ggplot(PollutionData, aes(x = precip, y = SO2, color = region)) +  
 geom\_point(alpha = 0.6) +   
 labs(title = "Precip and Region on SO2",   
 x = "Precip",   
 y = "SO2") +  
 theme\_minimal()  
r4 <- ggplot(PollutionData, aes(x = predays, y = SO2, color = region)) +  
 geom\_point(alpha = 0.6) +   
 labs(title = "Predays and Region on SO2",   
 x = "Predays",   
 y = "SO2") +  
 theme\_minimal()+   
 theme(legend.position = "none")  
r5 <- ggplot(PollutionData, aes(x = temp, y = SO2, color = region)) +  
 geom\_point(alpha = 0.6) +   
 labs(title = "Temp and Region on SO2",   
 x = "Temp",   
 y = "SO2") +  
 theme\_minimal()+   
 theme(legend.position = "none")  
r6 <- ggplot(PollutionData, aes(x = wind, y = SO2, color = region)) +  
 geom\_point(alpha = 0.6) +   
 labs(title = "Wind and Region on SO2",   
 x = "Wind",   
 y = "SO2") +  
 theme\_minimal()  
  
r1 + r2 + r3



r4 +r5 + r6



#### A2.6.2 Correlation of Categorical Variable

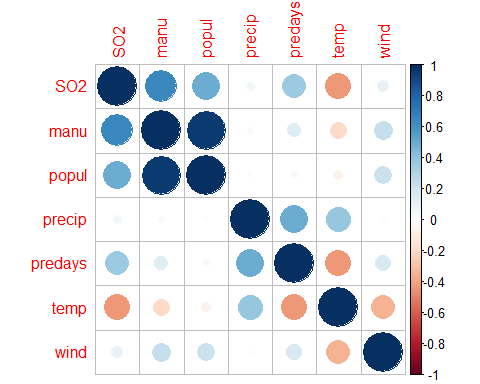
#### A2.6.3 Transformation of Categorical Variables

PD\_w\_dummies <- PollutionData %>%  
 mutate(  
 region\_East = ifelse(region == "East", 1, 0),  
 region\_North = ifelse(region == "North", 1, 0),  
 region\_South = ifelse(region == "South", 1, 0),  
 region\_West = ifelse(region == "West", 1, 0)  
 )  
#PD\_w\_dummies <- PD\_w\_dummies %>% select(-region)  
# View the new dataset  
tail(PD\_w\_dummies)

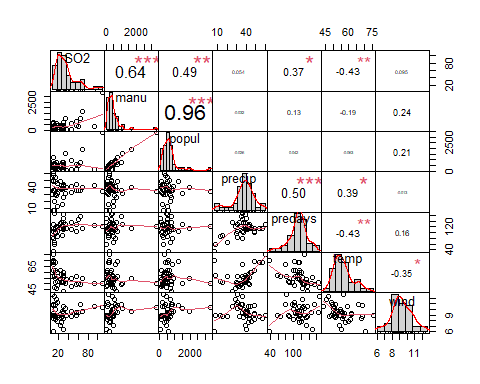
SO2 manu popul precip predays temp wind region ID region\_East region\_North  
36 12 453 716 20.66 67 56.7 8.7 West 36 0 0  
37 29 379 531 38.79 164 51.1 9.4 West 37 0 0  
38 56 775 622 35.89 105 55.9 9.5 West 38 0 0  
39 29 434 757 38.89 111 57.3 9.3 West 39 0 0  
40 8 125 277 30.58 82 56.6 12.7 North 40 0 1  
41 36 80 80 40.25 114 54.0 9.0 South 41 0 0  
 region\_South region\_West  
36 0 1  
37 0 1  
38 0 1  
39 0 1  
40 0 0  
41 1 0

### A2.7 Further Exploration for MLR specifically

#correlation matrix - large circles instead of just numbers  
PD\_Num <- PollutionData %>% select(-region, -ID)  
corrplot(cor(PD\_Num))



suppressWarnings(chart.Correlation(PD\_Num))



PD\_multi.lm <- lm(SO2 ~ manu + popul + precip + predays + temp + wind + region, data=PollutionData)  
#PD\_multi.lm <- lm(SO2 ~ popul + precip + predays + temp + wind + region, data=PollutionData)  
#PD\_multi.lm <- lm(SO2 ~ manu + precip + predays + temp + wind + region, data=PollutionData)  
  
  
vif(PD\_multi.lm)

GVIF Df GVIF^(1/(2\*Df))  
manu 14.756778 1 3.841455  
popul 14.524371 1 3.811085  
precip 4.473424 1 2.115047  
predays 3.782431 1 1.944847  
temp 4.819684 1 2.195378  
wind 1.374699 1 1.172476  
region 2.699925 3 1.180027

### A2.8 Results of Exploratory Analysis

## A3. Simple Linear Regression

### A3.1 Model Postulation - Pre-Transformation

#### A3.1.1 Parameter Estimation

# Variables fitted: manu, predays, temp   
  
cat("##### Line of Best Fit for 'SO2' and 'manu' \n")

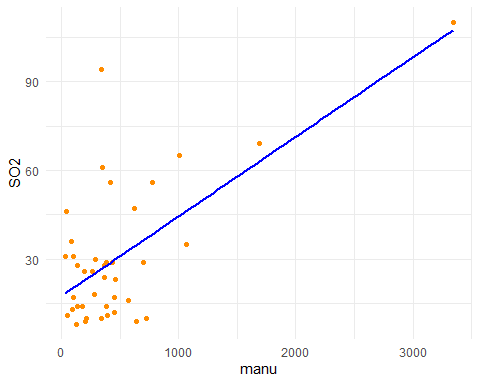
##### Line of Best Fit for 'SO2' and 'manu'

manu\_lm <- fitline\_function('manu', PollutionData)  
manu\_summary <- lmsummary\_function('manu', manu\_lm)  
manu\_summary

Call:  
lm(formula = formula, data = data)  
  
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

manu\_lmplot <- lmplot\_function('manu','darkorange', PollutionData)

Warning: `aes\_string()` was deprecated in ggplot2 3.0.0.  
ℹ Please use tidy evaluation idioms with `aes()`.  
ℹ See also `vignette("ggplot2-in-packages")` for more information.  
This warning is displayed once every 8 hours.  
Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
generated.



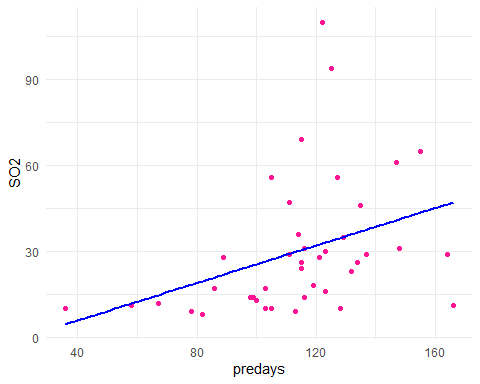
cat("##### Line of Best Fit for 'SO2' and 'predays' \n")

##### Line of Best Fit for 'SO2' and 'predays'

predays\_lm <- fitline\_function('predays', PollutionData)  
predays\_summary <- lmsummary\_function('predays', predays\_lm)  
predays\_summary

Call:  
lm(formula = formula, data = data)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-36.098 -12.499 -4.408 5.919 77.301   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) -7.2270 15.3991 -0.469 0.6415   
predays 0.3273 0.1318 2.484 0.0174 \*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 22.09 on 39 degrees of freedom  
Multiple R-squared: 0.1366, Adjusted R-squared: 0.1144   
F-statistic: 6.169 on 1 and 39 DF, p-value: 0.0174

predays\_lmplot <- lmplot\_function('predays', 'deeppink1', PollutionData)



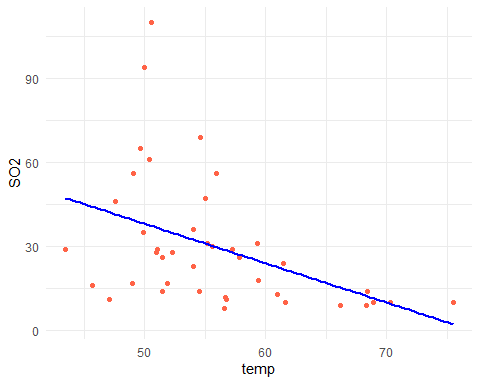
cat("##### Line of Best Fit for 'SO2' and 'temp' \n")

##### Line of Best Fit for 'SO2' and 'temp'

temp\_lm <- fitline\_function('temp', PollutionData)  
temp\_summary <- lmsummary\_function('temp', temp\_lm)  
temp\_summary

Call:  
lm(formula = formula, data = data)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-31.248 -11.830 -3.305 4.456 72.680   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 108.5711 26.3437 4.121 0.00019 \*\*\*  
temp -1.4081 0.4686 -3.005 0.00462 \*\*   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 21.42 on 39 degrees of freedom  
Multiple R-squared: 0.188, Adjusted R-squared: 0.1672   
F-statistic: 9.03 on 1 and 39 DF, p-value: 0.004624

temp\_lmplot <- lmplot\_function('temp', 'tomato', PollutionData)



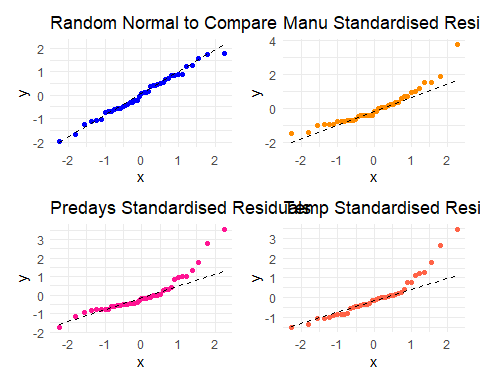
#### A3.1.2 Model Checking

##### A3.1.2.1 Checking SLR assumption with Residuals

# Get the residuals from the model  
manu\_residuals <- residuals(manu\_lm)  
predays\_residuals <- residuals(predays\_lm)  
temp\_residuals <- residuals(temp\_lm)  
  
# Standardise Residuals  
manu\_std.residuals <- rstandard(manu\_lm)  
predays\_std.residuals <- rstandard(predays\_lm)  
temp\_std.residuals <- rstandard(temp\_lm)

###### A3.1.2.1.1 Normality Check

##QQ PLots  
  
  
# Create some random normal data to compare  
set.seed(123)   
n <- length(manu\_std.residuals)   
random\_normal <- rnorm(n)  
  
  
random\_qqplot <- ggplot\_qq(random\_normal, "Random Normal to Compare", "blue")  
manu\_qqplot <- ggplot\_qq(manu\_std.residuals, "Manu Standardised Residuals", "darkorange")  
predays\_qqplot <- ggplot\_qq(predays\_std.residuals, "Predays Standardised Residuals", "deeppink1")  
temp\_qqplot <- ggplot\_qq(temp\_std.residuals, "Temp Standardised Residuals", "tomato")  
  
  
combined\_qqplot <- (random\_qqplot + manu\_qqplot) / (predays\_qqplot + temp\_qqplot)  
combined\_qqplot

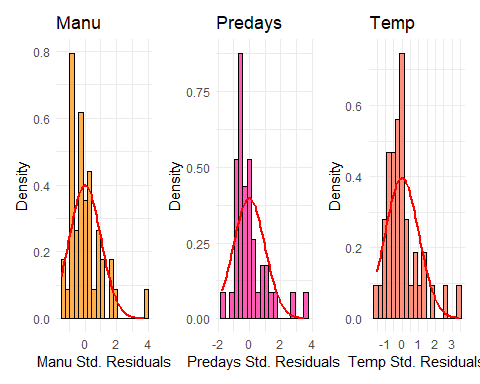


## Histograms  
  
# Create ggplot histograms  
manu\_hist\_plot <- ggplot\_hist\_with\_normal(manu\_std.residuals, "Manu", "Manu Std. Residuals", "darkorange")

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
ℹ Please use `linewidth` instead.  
This warning is displayed once every 8 hours.  
Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
generated.

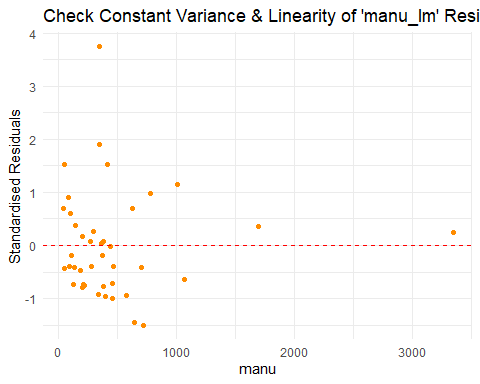
predays\_hist\_plot <- ggplot\_hist\_with\_normal(predays\_std.residuals, "Predays", "Predays Std. Residuals", "deeppink1")  
temp\_hist\_plot <- ggplot\_hist\_with\_normal(temp\_std.residuals, "Temp", "Temp Std. Residuals", "tomato")  
  
manu\_hist\_plot + predays\_hist\_plot + temp\_hist\_plot

Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.  
ℹ Please use `after\_stat(density)` instead.  
This warning is displayed once every 8 hours.  
Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
generated.

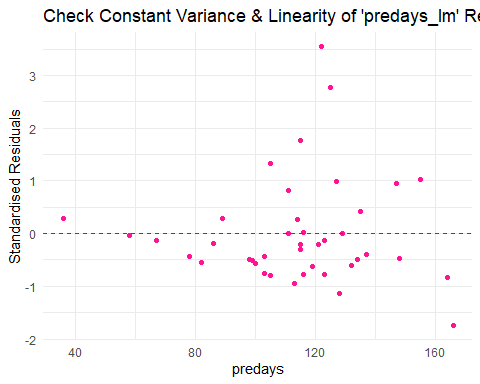


###### A3.1.2.1.2 Constant Variance & Linearity

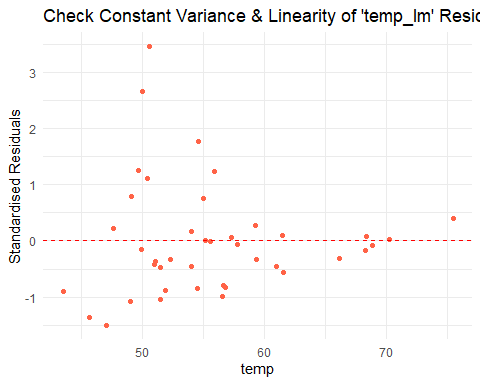
ggplot(PollutionData, aes(x = manu, y = manu\_std.residuals)) +  
 geom\_point(color = 'darkorange') +   
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +   
 labs(x = "manu", y = "Standardised Residuals",   
 title = "Check Constant Variance & Linearity of 'manu\_lm' Residuals") +  
 theme\_minimal()



ggplot(PollutionData, aes(x = predays, y = predays\_std.residuals)) +  
 geom\_point(color = 'deeppink1') +   
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +   
 labs(x = "predays", y = "Standardised Residuals",   
 title = "Check Constant Variance & Linearity of 'predays\_lm' Residuals") +  
 theme\_minimal()

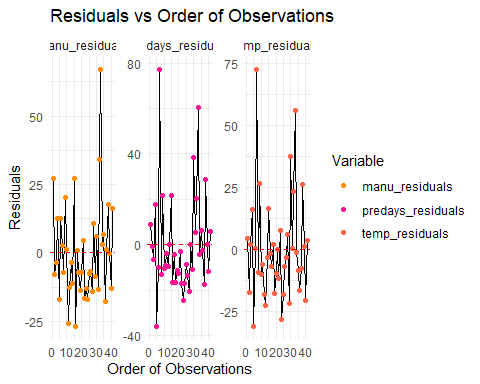


ggplot(PollutionData, aes(x = temp, y = temp\_std.residuals)) +  
 geom\_point(color = 'tomato') +   
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +   
 labs(x = "temp", y = "Standardised Residuals",   
 title = "Check Constant Variance & Linearity of 'temp\_lm' Residuals") +  
 theme\_minimal()



###### A3.1.2.1.3 Residual Independence

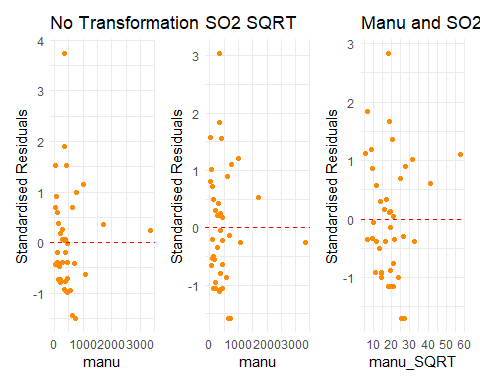
Residualdf <- data.frame(  
 manu\_residuals = manu\_residuals,  
 predays\_residuals = predays\_residuals,  
 temp\_residuals = temp\_residuals  
)  
Residualdf$order <- 1:nrow(Residualdf)  
  
  
# Create a new data frame with all residuals  
Residualdf\_long <- Residualdf %>%  
 pivot\_longer(cols = c(manu\_residuals, predays\_residuals, temp\_residuals),   
 names\_to = "Variable",   
 values\_to = "Residuals")  
  
# Create the plot  
ggplot(Residualdf\_long, aes(x = order, y = Residuals, color = Variable)) +  
 geom\_line(color = "black") + # Line plot for residuals  
 geom\_point() + # Points for residuals  
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") + # Horizontal line at y = 0  
 labs(x = "Order of Observations", y = "Residuals",   
 title = "Residuals vs Order of Observations") +  
 theme\_minimal() +  
 scale\_color\_manual(values = c("manu\_residuals" = "darkorange",   
 "predays\_residuals" = "deeppink1",   
 "temp\_residuals" = "tomato")) +  
 facet\_wrap(~ Variable, scales = "free") # Separate plots for each residuals group



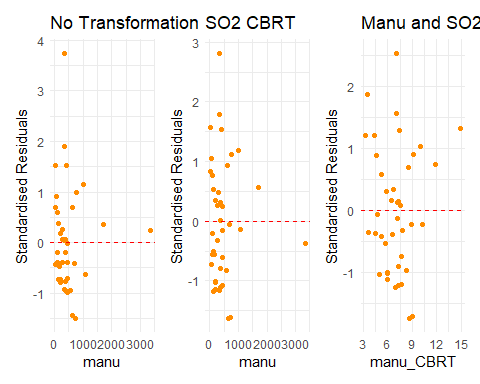
### A3.2 Variable Transformation

#### A3.2.1 Variable Transformation Assessment

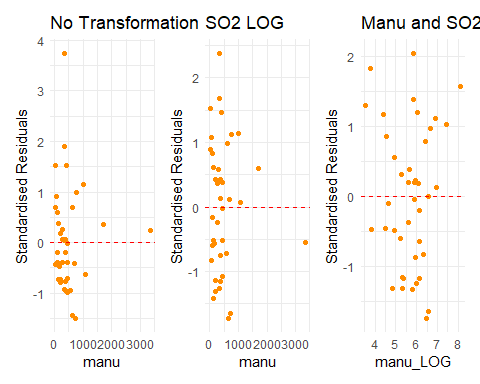
PD\_Trans <- PollutionData  
#View(PD\_Trans)  
  
#Sqrt  
PD\_Trans$SO2\_SQRT <- sqrt(PollutionData$SO2)  
PD\_Trans$manu\_SQRT <- sqrt(PollutionData$manu)  
PD\_Trans$predays\_SQRT <- sqrt(PollutionData$predays)  
PD\_Trans$temp\_SQRT <- sqrt(PollutionData$temp)  
  
#Cbrt  
PD\_Trans$SO2\_CBRT <- (PollutionData$SO2)^(1/3)  
PD\_Trans$manu\_CBRT <- (PollutionData$manu)^(1/3)  
PD\_Trans$predays\_CBRT <- (PollutionData$predays)^(1/3)  
PD\_Trans$temp\_CBRT <- (PollutionData$temp)^(1/3)  
  
#log  
PD\_Trans$SO2\_LOG <- log(PollutionData$SO2)  
PD\_Trans$manu\_LOG <- log(PollutionData$manu)  
PD\_Trans$predays\_LOG <- log(PollutionData$predays)  
PD\_Trans$temp\_LOG <- log(PollutionData$temp)  
  
  
  
##manu  
  
#SQRT  
manu\_notrans\_plot <- residual\_plot\_function(PD\_Trans, "manu", "SO2", "No Transformation", "darkorange")  
manu\_SO2SQRT\_plot <- residual\_plot\_function(PD\_Trans, "manu", "SO2\_SQRT", "SO2 SQRT", "darkorange")  
manu\_2SQRT\_plot <- residual\_plot\_function(PD\_Trans, "manu\_SQRT", "SO2\_SQRT", "Manu and SO2 SQRT", "darkorange")  
  
#SQRT Comparison  
manu\_notrans\_plot + manu\_SO2SQRT\_plot + manu\_2SQRT\_plot



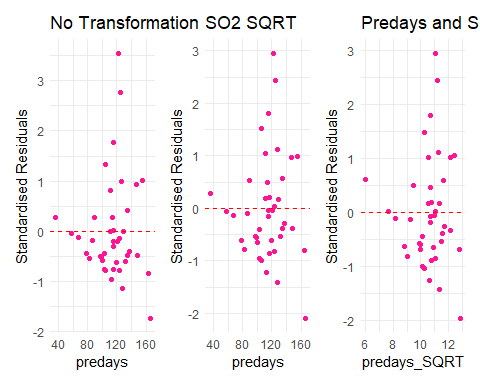
#CBRT  
manu\_SO2CBRT\_plot <- residual\_plot\_function(PD\_Trans, "manu", "SO2\_CBRT", "SO2 CBRT", "darkorange")  
manu\_2CBRT\_plot <- residual\_plot\_function(PD\_Trans, "manu\_CBRT", "SO2\_CBRT", "Manu and SO2 CBRT", "darkorange")  
  
#CBRT Comparison  
manu\_notrans\_plot + manu\_SO2CBRT\_plot + manu\_2CBRT\_plot



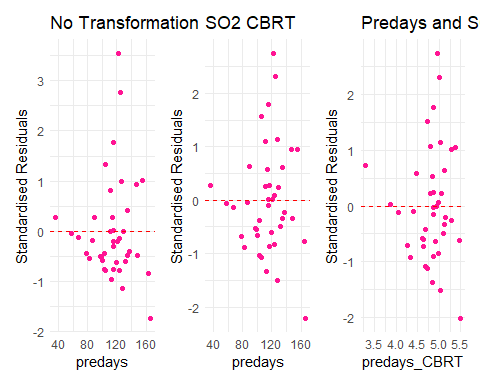
#LOG  
manu\_SO2LOG\_plot <- residual\_plot\_function(PD\_Trans, "manu", "SO2\_LOG", "SO2 LOG", "darkorange")  
manu\_2LOG\_plot <- residual\_plot\_function(PD\_Trans, "manu\_LOG", "SO2\_LOG", "Manu and SO2 LOG", "darkorange")  
  
#CBRT Comparison  
manu\_notrans\_plot + manu\_SO2LOG\_plot + manu\_2LOG\_plot



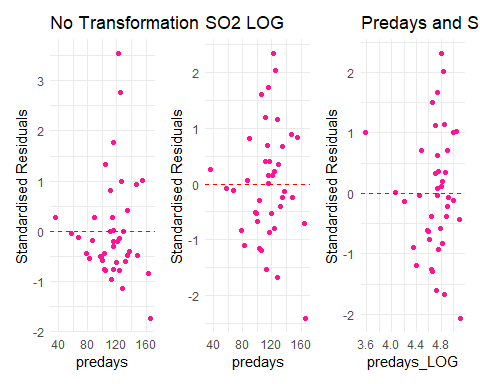
## predays  
  
# SQRT  
predays\_notrans\_plot <- residual\_plot\_function(PD\_Trans, "predays", "SO2", "No Transformation", "deeppink1")  
predays\_SO2SQRT\_plot <- residual\_plot\_function(PD\_Trans, "predays", "SO2\_SQRT", "SO2 SQRT", "deeppink1")  
predays\_2SQRT\_plot <- residual\_plot\_function(PD\_Trans, "predays\_SQRT", "SO2\_SQRT", "Predays and SO2 SQRT", "deeppink1")  
  
# SQRT Comparison  
predays\_notrans\_plot + predays\_SO2SQRT\_plot + predays\_2SQRT\_plot



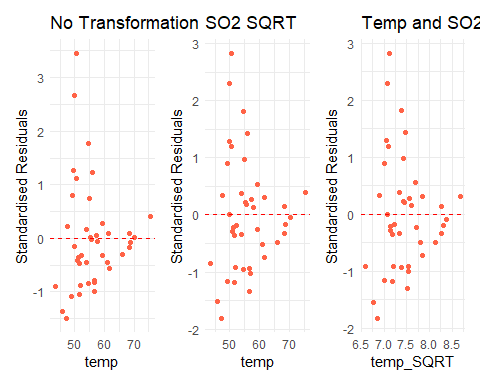
# CBRT  
predays\_SO2CBRT\_plot <- residual\_plot\_function(PD\_Trans, "predays", "SO2\_CBRT", "SO2 CBRT", "deeppink1")  
predays\_2CBRT\_plot <- residual\_plot\_function(PD\_Trans, "predays\_CBRT", "SO2\_CBRT", "Predays and SO2 CBRT", "deeppink1")  
  
# CBRT Comparison  
predays\_notrans\_plot + predays\_SO2CBRT\_plot + predays\_2CBRT\_plot



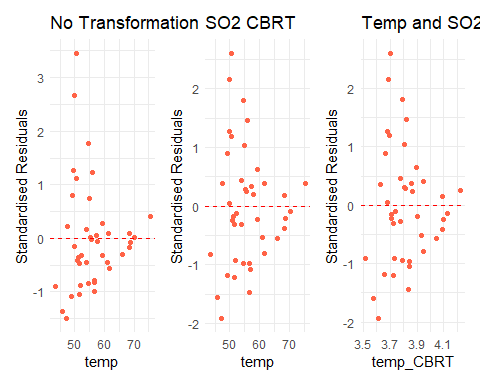
# LOG  
predays\_SO2LOG\_plot <- residual\_plot\_function(PD\_Trans, "predays", "SO2\_LOG", "SO2 LOG", "deeppink1")  
predays\_2LOG\_plot <- residual\_plot\_function(PD\_Trans, "predays\_LOG", "SO2\_LOG", "Predays and SO2 LOG", "deeppink1")  
  
# LOG Comparison  
predays\_notrans\_plot + predays\_SO2LOG\_plot + predays\_2LOG\_plot



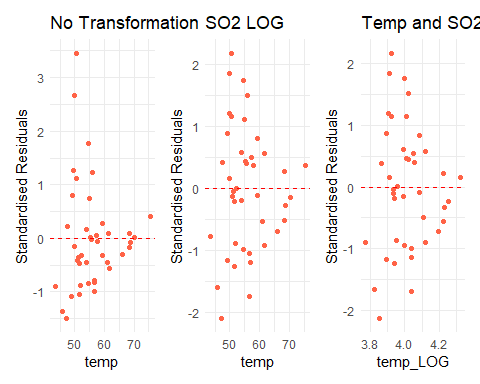
## temp  
  
# SQRT  
temp\_notrans\_plot <- residual\_plot\_function(PD\_Trans, "temp", "SO2", "No Transformation", "tomato")  
temp\_SO2SQRT\_plot <- residual\_plot\_function(PD\_Trans, "temp", "SO2\_SQRT", "SO2 SQRT", "tomato")  
temp\_2SQRT\_plot <- residual\_plot\_function(PD\_Trans, "temp\_SQRT", "SO2\_SQRT", "Temp and SO2 SQRT", "tomato")  
  
# SQRT Comparison  
temp\_notrans\_plot + temp\_SO2SQRT\_plot + temp\_2SQRT\_plot



# CBRT  
temp\_SO2CBRT\_plot <- residual\_plot\_function(PD\_Trans, "temp", "SO2\_CBRT", "SO2 CBRT", "tomato")  
temp\_2CBRT\_plot <- residual\_plot\_function(PD\_Trans, "temp\_CBRT", "SO2\_CBRT", "Temp and SO2 CBRT", "tomato")  
  
# CBRT Comparison  
temp\_notrans\_plot + temp\_SO2CBRT\_plot + temp\_2CBRT\_plot



# LOG  
temp\_SO2LOG\_plot <- residual\_plot\_function(PD\_Trans, "temp", "SO2\_LOG", "SO2 LOG", "tomato")  
temp\_2LOG\_plot <- residual\_plot\_function(PD\_Trans, "temp\_LOG", "SO2\_LOG", "Temp and SO2 LOG", "tomato")  
  
# LOG Comparison  
temp\_notrans\_plot + temp\_SO2LOG\_plot + temp\_2LOG\_plot



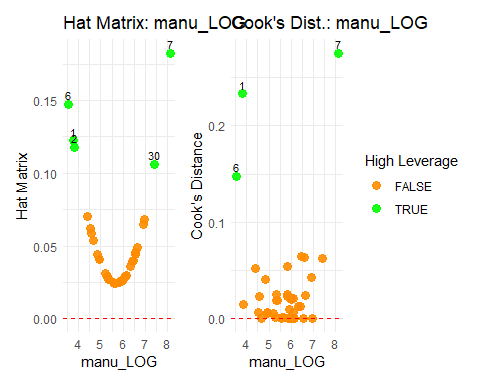
#### A3.2.2 Dataframe Transformation

# manu data frame  
  
PD\_logmanu\_logSO2 <- data.frame(  
 ID = PD\_Trans$ID,  
 SO2 = PD\_Trans$SO2,  
 SO2\_LOG = PD\_Trans$SO2\_LOG,  
 manu = PD\_Trans$manu,  
 manu\_LOG = PD\_Trans$manu\_LOG,  
 stringsAsFactors = FALSE  
)  
  
# predays data frame  
  
PD\_predays\_logSO2 <- data.frame(  
 ID = PD\_Trans$ID,  
 SO2 = PD\_Trans$SO2,  
 SO2\_LOG = PD\_Trans$SO2\_LOG,  
 predays = PD\_Trans$predays,  
 stringsAsFactors = FALSE  
)

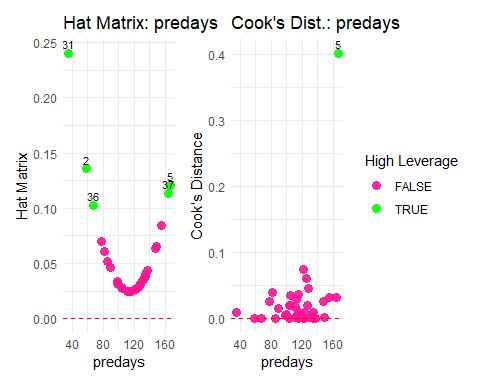
### A3.3 Outlier Influence Analysis

#### A3.3.1 Calculate Hat Matrix Element and Cooks Distance

manu\_hat\_values <- plot\_hat\_values(PD\_logmanu\_logSO2, "manu\_LOG", "SO2\_LOG", false\_color = "darkorange")  
predays\_hat\_values <- plot\_hat\_values(PD\_predays\_logSO2, "predays", "SO2\_LOG", false\_color = "deeppink1")  
#plot\_hat\_values(PollutionData, "temp", false\_color = "tomato")   
  
manu\_cooks\_distance <- plot\_cooks\_distance(PD\_logmanu\_logSO2, "manu\_LOG", "SO2\_LOG", false\_color = "darkorange")   
predays\_cooks\_distance <- plot\_cooks\_distance(PD\_predays\_logSO2, "predays", "SO2\_LOG", false\_color = "deeppink1")   
#plot\_cooks\_distance(PollutionData, "temp", false\_color = "tomato")  
  
manu\_hat\_values + manu\_cooks\_distance

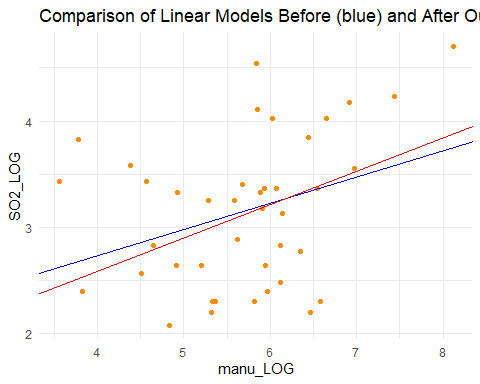


predays\_hat\_values + predays\_cooks\_distance



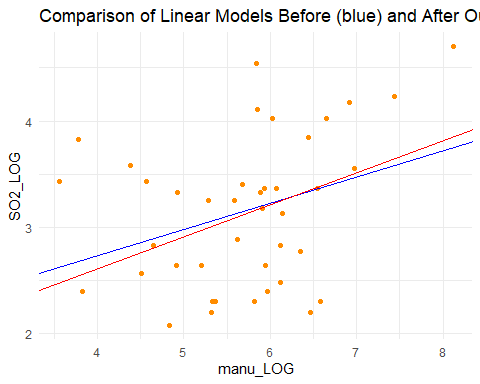
#### A3.3.2 Assess Leverage

compare\_lm\_function(PD\_logmanu\_logSO2, "manu\_LOG", "SO2\_LOG", 1, "darkorange")



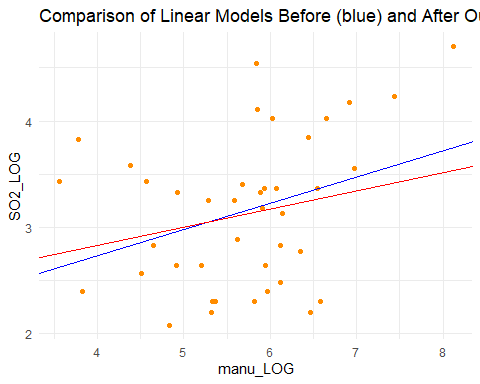
##### Compare ANOVA Tables for manu\_LOG Outlier: ID 1  
Pre-Outlier-Removal:  
MSE: 0.4480219  
Post-Outlier-Removal:  
MSE: 0.4204332  
  
Model after Outlier Removal fits better (due to lower MSE = higher F = lower pvalue).

compare\_lm\_function(PD\_logmanu\_logSO2, "manu\_LOG", "SO2\_LOG", 6, "darkorange")



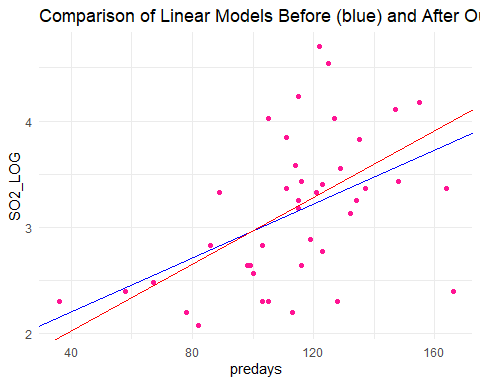
##### Compare ANOVA Tables for manu\_LOG Outlier: ID 6  
Pre-Outlier-Removal:  
MSE: 0.4480219  
Post-Outlier-Removal:  
MSE: 0.439677  
  
Model after Outlier Removal fits better (due to lower MSE = higher F = lower pvalue).

compare\_lm\_function(PD\_logmanu\_logSO2, "manu\_LOG", "SO2\_LOG", 7, "darkorange")



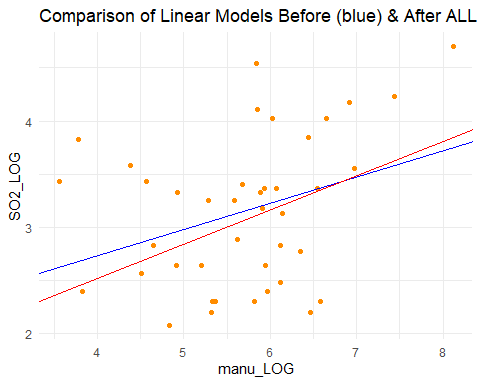
##### Compare ANOVA Tables for manu\_LOG Outlier: ID 7  
Pre-Outlier-Removal:  
MSE: 0.4480219  
Post-Outlier-Removal:  
MSE: 0.4307456  
  
Model after Outlier Removal fits better (due to lower MSE = higher F = lower pvalue).

compare\_lm\_function(PD\_predays\_logSO2, "predays", "SO2\_LOG", 5, "deeppink1")



##### Compare ANOVA Tables for predays Outlier: ID 5  
Pre-Outlier-Removal:  
MSE: 0.3900124  
Post-Outlier-Removal:  
MSE: 0.3402759  
  
Model after Outlier Removal fits better (due to lower MSE = higher F = lower pvalue).

#Does removing all outliers in manu allow for a better model than removing none?  
initial\_model <- lm(SO2\_LOG ~ manu\_LOG, data = PD\_logmanu\_logSO2)  
   
#Remove the specified rows  
data\_no\_outliers <- PD\_logmanu\_logSO2[-c(1, 6, 7), ]  
  
#Fit the linear model after removing the row  
model\_no\_outlier <- lm(SO2\_LOG ~ manu\_LOG, data\_no\_outliers)  
  
  
#Plot both linear models  
plot <- ggplot(PD\_logmanu\_logSO2, aes(x = manu\_LOG, y = SO2\_LOG)) +  
 geom\_point(color = "darkorange") +  
 geom\_abline(intercept = coef(initial\_model)[1],   
 slope = coef(initial\_model)[2],   
 color = "blue") +  
 geom\_abline(intercept = coef(model\_no\_outlier)[1],   
 slope = coef(model\_no\_outlier)[2],   
 color = "red") +  
   
 theme\_minimal() +  
 labs(title = "Comparison of Linear Models Before (blue) & After ALL Outlier Removal (red)",  
 x = "manu\_LOG",  
 y = "SO2\_LOG")  
  
print(plot)



# Step 5: Calculate ANOVA and print results  
anova\_initial <- anova(initial\_model)  
anova\_no\_outlier <- anova(model\_no\_outlier)  
  
  
mse\_initial <- anova\_initial["Residuals", "Mean Sq"]  
mse\_no\_outlier <- anova\_no\_outlier["Residuals", "Mean Sq"]  
  
cat("\n##### Compare ANOVA Tables for manu\_LOG. Outlier: ID's 1, 6, 7")

##### Compare ANOVA Tables for manu\_LOG. Outlier: ID's 1, 6, 7

cat("\nPre-Outlier-Removal:")

Pre-Outlier-Removal:

cat("\nMSE:", mse\_initial)

MSE: 0.4480219

cat("\nPost-Outlier-Removal:")

Post-Outlier-Removal:

cat("\nMSE:", mse\_no\_outlier)

MSE: 0.3958783

# Step 6: Determine which model fits better  
if (mse\_no\_outlier < mse\_initial) {  
 cat("\n\nModel after Outlier Removal fits better (due to lower MSE = higher F = lower pvalue).\n")  
} else {  
 cat("\n\nModel before Outlier Removal fits better (due to lower MSE = higher F = lower pvalue).\n")  
}

Model after Outlier Removal fits better (due to lower MSE = higher F = lower pvalue).

#### A3.3.4 Outlier Removals

# Remove Outlier 5 from pre-days and make new dataframe  
  
PD\_predays\_logSO2\_r5 <- PD\_predays\_logSO2  
PD\_predays\_logSO2\_r5 <- PD\_predays\_logSO2\_r5[-5, ]  
  
  
# Remove Outlier 5 from pre-days and make new dataframe  
PD\_logmanu\_logSO2\_r167<- PD\_logmanu\_logSO2  
PD\_logmanu\_logSO2\_r167<- PD\_logmanu\_logSO2\_r167[-c(1, 6, 7), ]

### A3.4 Model Postulation - Post-Transformation & Outlier Removals

#### A3.4.1 Parameter Estimation

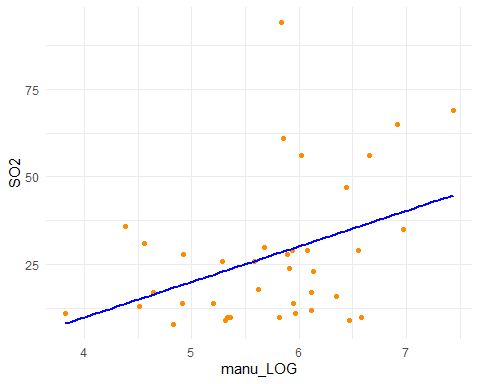
# Variables fitted: manu, predays, temp   
  
cat("##### Line of Best Fit for 'SO2\_LOG' and 'manu\_LOG' (post trans)\n")

##### Line of Best Fit for 'SO2\_LOG' and 'manu\_LOG' (post trans)

manu2\_lm <- lm(SO2\_LOG ~ manu\_LOG, data = PD\_logmanu\_logSO2\_r167)  
manu2\_summary <- lmsummary\_function('manu\_LOG', manu2\_lm)  
manu2\_summary

Call:  
lm(formula = SO2\_LOG ~ manu\_LOG, data = PD\_logmanu\_logSO2\_r167)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-1.1243 -0.5139 0.0265 0.4640 1.4230   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 1.2408 0.7595 1.634 0.1110   
manu\_LOG 0.3219 0.1312 2.453 0.0191 \*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.6292 on 36 degrees of freedom  
Multiple R-squared: 0.1432, Adjusted R-squared: 0.1194   
F-statistic: 6.017 on 1 and 36 DF, p-value: 0.01914

manu2\_lmplot <- lmplot\_function('manu\_LOG','darkorange', PD\_logmanu\_logSO2\_r167)



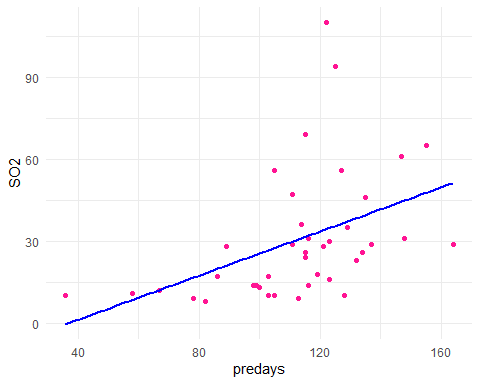
cat("##### Line of Best Fit for 'SO2\_LOG' and 'predays' (post trans) \n")

##### Line of Best Fit for 'SO2\_LOG' and 'predays' (post trans)

predays2\_lm <- lm(SO2\_LOG ~ predays, PD\_predays\_logSO2\_r5)  
predays2\_summary <- lmsummary\_function('predays', predays2\_lm)  
predays2\_summary

Call:  
lm(formula = SO2\_LOG ~ predays, data = PD\_predays\_logSO2\_r5)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-1.11054 -0.38871 0.02805 0.33258 1.38134   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 1.407963 0.422949 3.329 0.001946 \*\*   
predays 0.015665 0.003666 4.273 0.000124 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.5833 on 38 degrees of freedom  
Multiple R-squared: 0.3246, Adjusted R-squared: 0.3068   
F-statistic: 18.26 on 1 and 38 DF, p-value: 0.0001245

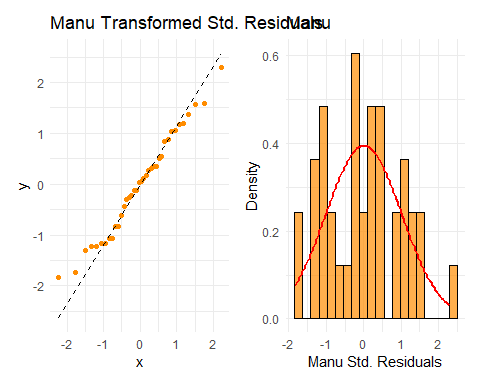
predays2\_lmplot <- lmplot\_function('predays', 'deeppink1', PD\_predays\_logSO2\_r5)



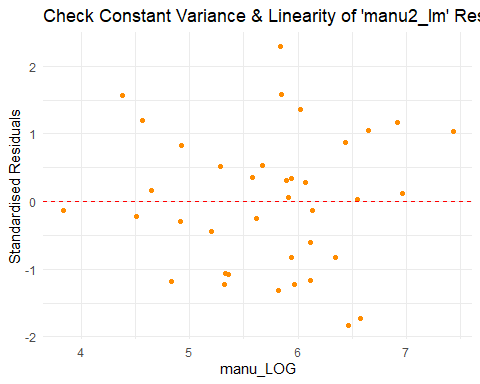
#### A3.4.2 Model Checking

##### A3.4.2.1 Checking SLR assumption with Residuals

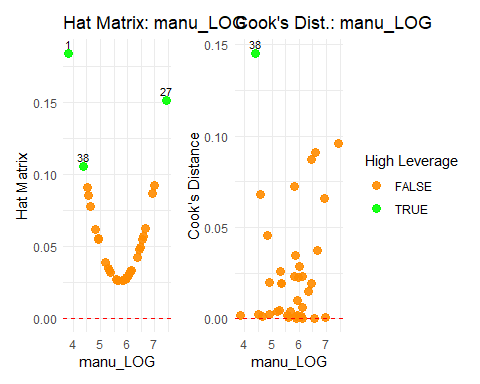
#manu   
# Get the residuals from the model  
manu2\_residuals <- residuals(manu2\_lm)  
  
# Standardise Residuals  
manu2\_std.residuals <- rstandard(manu2\_lm)  
  
#Check Residuals  
manu2\_qqplot <- ggplot\_qq(manu2\_std.residuals, "Manu Transformed Std. Residuals", "darkorange")  
manu2\_hist\_plot <- ggplot\_hist\_with\_normal(manu2\_std.residuals, "Manu", "Manu Std. Residuals", "darkorange")  
  
manu2\_resplot <- ggplot(PD\_logmanu\_logSO2\_r167, aes(x = manu\_LOG, y = manu2\_std.residuals)) +  
 geom\_point(color = 'darkorange') +   
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +   
 labs(x = "manu\_LOG", y = "Standardised Residuals",   
 title = "Check Constant Variance & Linearity of 'manu2\_lm' Residuals") +  
 theme\_minimal()   
  
#Check Outliers  
manu2\_hat\_values <- plot\_hat\_values(PD\_logmanu\_logSO2\_r167, "manu\_LOG", "SO2\_LOG", false\_color = "darkorange")  
manu2\_cooks\_distance <- plot\_cooks\_distance(PD\_logmanu\_logSO2\_r167, "manu\_LOG", "SO2\_LOG", false\_color = "darkorange")   
  
manu2\_qqplot +   
manu2\_hist\_plot



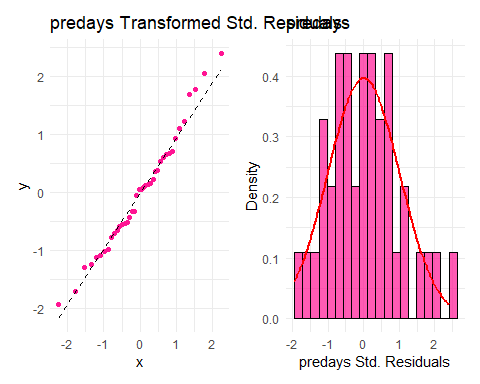
manu2\_resplot



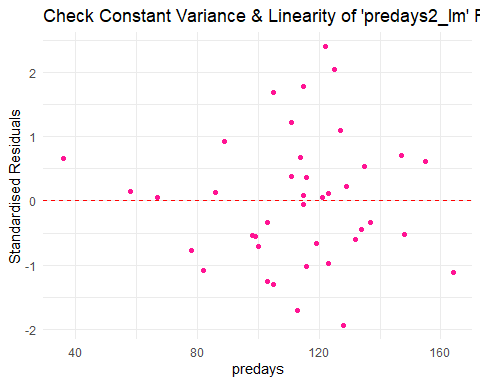
manu2\_hat\_values +  
manu2\_cooks\_distance



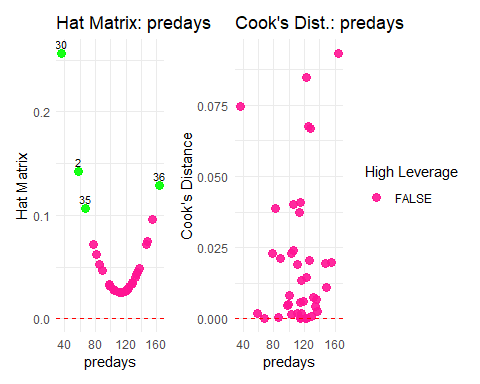
#predays   
# Get the residuals from the model  
predays2\_residuals <- residuals(predays2\_lm)  
  
# Standardise Residuals  
predays2\_std.residuals <- rstandard(predays2\_lm)  
  
#Check Residuals  
predays2\_qqplot <- ggplot\_qq(predays2\_std.residuals, "predays Transformed Std. Residuals", "deeppink1")  
predays2\_hist\_plot <- ggplot\_hist\_with\_normal(predays2\_std.residuals, "predays", "predays Std. Residuals", "deeppink1")  
  
predays2\_resplot <- ggplot(PD\_predays\_logSO2\_r5, aes(x = predays, y = predays2\_std.residuals)) +  
 geom\_point(color = 'deeppink1') +   
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +   
 labs(x = "predays", y = "Standardised Residuals",   
 title = "Check Constant Variance & Linearity of 'predays2\_lm' Residuals") +  
 theme\_minimal()   
  
#Check Outliers  
predays2\_hat\_values <- plot\_hat\_values(PD\_predays\_logSO2\_r5, "predays", "SO2\_LOG", false\_color = "deeppink1")  
predays2\_cooks\_distance <- plot\_cooks\_distance(PD\_predays\_logSO2\_r5, "predays", "SO2\_LOG", false\_color = "deeppink1")   
  
predays2\_qqplot +   
predays2\_hist\_plot



predays2\_resplot



predays2\_hat\_values +  
predays2\_cooks\_distance



### A3.5 Model Use

#### A3.5.1 ANOVA F Test

anova(manu2\_lm)

Analysis of Variance Table  
  
Response: SO2\_LOG  
 Df Sum Sq Mean Sq F value Pr(>F)   
manu\_LOG 1 2.3822 2.38218 6.0174 0.01914 \*  
Residuals 36 14.2516 0.39588   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

anova(predays2\_lm)

Analysis of Variance Table  
  
Response: SO2\_LOG  
 Df Sum Sq Mean Sq F value Pr(>F)   
predays 1 6.214 6.2140 18.262 0.0001245 \*\*\*  
Residuals 38 12.931 0.3403   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#### A3.5.2 Coefficient of Determination

model\_summary\_manu2 <- summary(manu2\_lm)  
model\_summary\_predays2 <- summary(predays2\_lm)  
  
# Extract R-squared value   
r\_squared\_manu2 <- model\_summary\_manu2$r.squared   
r\_squared\_predays2 <- model\_summary\_predays2$r.squared   
  
# Print R-squared value   
cat("For manu: The coefficient of determination (R^2) is:", r\_squared\_manu2, "\n")

For manu: The coefficient of determination (R^2) is: 0.143213

cat("For predays: The coefficient of determination (R^2) is:", r\_squared\_predays2, "\n")

For predays: The coefficient of determination (R^2) is: 0.3245845

#### A3.5.3 Confidence and Prediction Intervals

##### A3.5.3.1 Confidence and Prediction Intervals in Logarithmic Space

X <- PD\_logmanu\_logSO2\_r167$manu\_LOG  
Y <- PD\_logmanu\_logSO2\_r167$SO2\_LOG  
  
# ## Make the prediction X-values dataframe.  
X\_pred\_manu <- data.frame(manu\_LOG=seq(from=50, to=1000, by=50))  
X\_pred\_manu <- log(X\_pred\_manu)  
# #   
# # ## Make predictions, given the X values; that is, fit the model using all rows  
# # ## in the prediction X values. Here, we also target confidence intervals.  
# # ## NOTE See `?predict` for more details.  
Y\_pred\_conf\_manu <- predict(  
 manu2\_lm,  
 newdata=X\_pred\_manu,  
 interval="confidence",  
 level=0.95  
 )  
#   
# # ## Inspect the X and Y values.  
# X\_pred\_manu  
# #   
Y\_pred\_conf\_manu

fit lwr upr  
1 2.500280 1.972798 3.027762  
2 2.723436 2.358398 3.088474  
3 2.853973 2.571128 3.136819  
4 2.946591 2.709217 3.183965  
5 3.018431 2.803782 3.233080  
6 3.077129 2.869961 3.284296  
7 3.126757 2.917174 3.336340  
8 3.169747 2.951759 3.387734  
9 3.207666 2.977916 3.437417  
10 3.241587 2.998353 3.484820  
11 3.272271 3.014810 3.529732  
12 3.300284 3.028414 3.572155  
13 3.326054 3.039911 3.612196  
14 3.349912 3.049812 3.650013  
15 3.372124 3.058472 3.685777  
16 3.392902 3.066147 3.719657  
17 3.412420 3.073025 3.751814  
18 3.430822 3.079248 3.782395  
19 3.448228 3.084924 3.811533  
20 3.464742 3.090137 3.839347

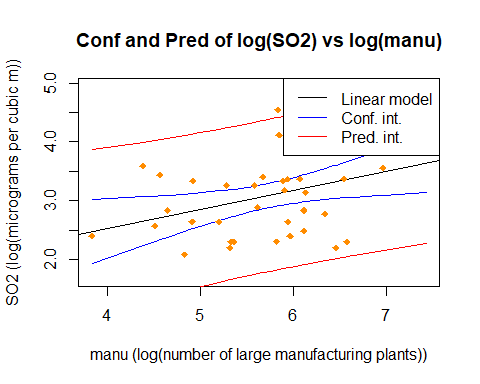
## Compute and inspect prediction intervals for "new X".  
Y\_pred\_pred\_manu <- predict(  
 manu2\_lm,  
 newdata=X\_pred\_manu,  
 interval="prediction",  
 level=0.95  
)  
  
Y\_pred\_pred\_manu

fit lwr upr  
1 2.500280 1.119501 3.881059  
2 2.723436 1.396196 4.050676  
3 2.853973 1.546948 4.160998  
4 2.946591 1.648647 4.244535  
5 3.018431 1.724450 4.312412  
6 3.077129 1.784368 4.369890  
7 3.126757 1.833607 4.419907  
8 3.169747 1.875208 4.464286  
9 3.207666 1.911095 4.504238  
10 3.241587 1.942558 4.540615  
11 3.272271 1.970504 4.574039  
12 3.300284 1.995590 4.604978  
13 3.326054 2.018311 4.633796  
14 3.349912 2.039045 4.660780  
15 3.372124 2.058088 4.686160  
16 3.392902 2.075677 4.710127  
17 3.412420 2.092003 4.732837  
18 3.430822 2.107222 4.754422  
19 3.448228 2.121464 4.774993  
20 3.464742 2.134839 4.794645

## - confidence and prediction intervals for \*observed\* X  
Y\_obs\_conf\_manu <- predict(manu2\_lm, interval="confidence", level=0.95)  
Y\_obs\_pred\_manu <- predict(manu2\_lm, interval="prediction", level=0.95)

Warning in predict.lm(manu2\_lm, interval = "prediction", level = 0.95): predictions on current data refer to \_future\_ responses

## - X ordering and ordered X  
X\_ordering <- order(X)  
X\_ordered <- X[X\_ordering]  
  
  
logspacemanu <- plot(  
 SO2\_LOG ~ manu\_LOG,  
 data=PD\_logmanu\_logSO2\_r167,  
 ylab="SO2 (log(micrograms per cubic m))",  
 xlab="manu (log(number of large manufacturing plants))",  
 main="Conf and Pred of log(SO2) vs log(manu)",  
 pch=".",   
 cex=0,   
 xlim = c(min(PD\_logmanu\_logSO2\_r167$manu\_LOG), max(PD\_logmanu\_logSO2\_r167$manu\_LOG)),  
 ylim = c(min(PD\_logmanu\_logSO2\_r167$SO2\_LOG) - 0.4, max(PD\_logmanu\_logSO2\_r167$SO2\_LOG) + 0.4)  
 )  
   
 abline(manu2\_lm, lwd=1.7)  
   
 matlines(  
 x=X\_ordered,  
 y=Y\_obs\_conf\_manu[X\_ordering, c(2, 3)],  
 col="blue",  
 lty=1,  
 lwd=1.7  
 )  
   
 matlines(  
 x=X\_ordered,  
 y=Y\_obs\_pred\_manu[X\_ordering, c(2, 3)],  
 col="red",  
 lty=1,  
 lwd=1.7  
 )  
   
 points(  
 SO2\_LOG ~ manu\_LOG,  
 data=PD\_logmanu\_logSO2\_r167,  
 pch=18L,  
 cex=1,   
 col='darkorange'  
 )  
   
 legend(  
 "topright",  
 legend = c("Linear model", "Conf. int.", "Pred. int."),  
 col = c("black", "blue", "red"),  
 lty = 1,  
 lwd = 1.8  
 )



X <- PD\_predays\_logSO2\_r5$predays  
Y <- PD\_predays\_logSO2\_r5$SO2\_LOG  
  
  
X\_pred\_predays <- data.frame(predays=seq(from=40, to=170, by=15))  
  
  
Y\_pred\_conf\_predays <- predict(  
 predays2\_lm,  
 newdata=X\_pred\_predays,  
 interval="confidence",  
 level=0.95  
 )  
  
Y\_pred\_conf\_predays

fit lwr upr  
1 2.034577 1.464372 2.604782  
2 2.269557 1.803104 2.736010  
3 2.504537 2.137379 2.871695  
4 2.739518 2.462364 3.016672  
5 2.974498 2.765678 3.183318  
6 3.209478 3.021915 3.397041  
7 3.444458 3.217442 3.671475  
8 3.679439 3.375013 3.983864  
9 3.914419 3.516178 4.312660

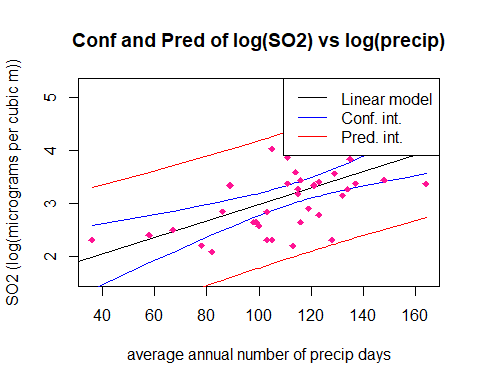
Y\_pred\_pred\_predays <- predict(  
 predays2\_lm,  
 newdata=X\_pred\_predays,  
 interval="prediction",  
 level=0.95  
)  
  
Y\_pred\_pred\_predays

fit lwr upr  
1 2.034577 0.7232254 3.345928  
2 2.269557 0.9998771 3.539237  
3 2.504537 1.2678830 3.741192  
4 2.739518 1.5265365 3.952499  
5 2.974498 1.7752836 4.173712  
6 3.209478 2.0137822 4.405174  
7 3.444458 2.2419423 4.646975  
8 3.679439 2.4599372 4.898940  
9 3.914419 2.6681827 5.160655

Y\_obs\_conf\_predays <- predict(predays2\_lm, interval="confidence", level=0.95)  
Y\_obs\_pred\_predays <- predict(predays2\_lm, interval="prediction", level=0.95)

Warning in predict.lm(predays2\_lm, interval = "prediction", level = 0.95): predictions on current data refer to \_future\_ responses

X\_ordering <- order(X)  
X\_ordered <- X[X\_ordering]  
  
## Make the base scatterplot.  
logspacepredays <- plot(  
 SO2\_LOG ~ predays,  
 data=PD\_predays\_logSO2\_r5,  
 ylab="SO2 (log(micrograms per cubic m))",  
 xlab="average annual number of precip days",  
 main="Conf and Pred of log(SO2) vs log(precip)",  
 pch=".",   
 cex=0,   
 ylim = c(min(PD\_predays\_logSO2\_r5$SO2\_LOG) - 0.5, max(PD\_predays\_logSO2\_r5$SO2\_LOG) + 0.5)  
 )  
  
 abline(predays2\_lm, lwd=1.7)  
   
  
 matlines(  
 x=X\_ordered,  
 y=Y\_obs\_conf\_predays[X\_ordering, c(2, 3)],  
 col="blue",  
 lty=1,  
 lwd=1.7  
 )  
  
 matlines(  
 x=X\_ordered,  
 y=Y\_obs\_pred\_predays[X\_ordering, c(2, 3)],  
 col="red",  
 lty=1,  
 lwd=1.7  
 )  
   
 points(  
 SO2\_LOG ~ predays,  
 data=PD\_predays\_logSO2\_r5,  
 pch=18L,  
 cex=1,   
 col='deeppink1'  
 )  
   
 legend(  
 "topright",  
 legend = c("Linear model", "Conf. int.", "Pred. int."),  
 col = c("black", "blue", "red"),  
 lty = 1,  
 lwd = 1.8  
 )

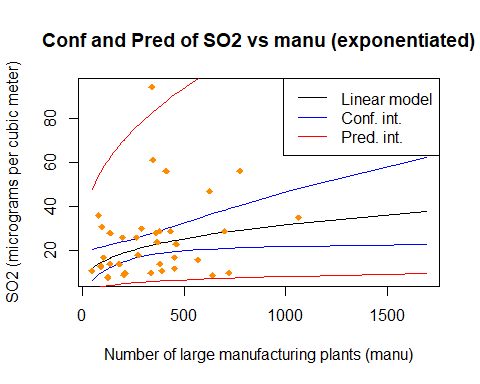


##### A3.5.3.2 Confidence and Prediction Intervals in Exponentiated Space

# Get confidence and prediction intervals for the linear model in log space  
manu2\_conf <- predict(manu2\_lm, interval = "conf")  
manu2\_pred <- predict(manu2\_lm, interval = "pred")

Warning in predict.lm(manu2\_lm, interval = "pred"): predictions on current data refer to \_future\_ responses

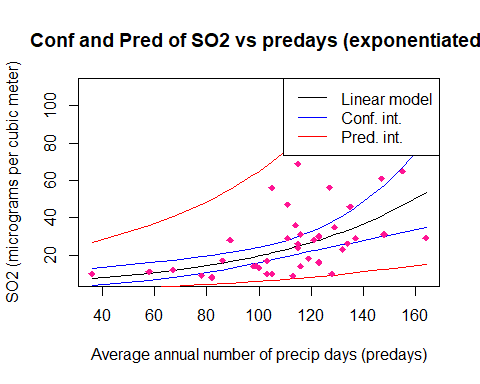
# Rename columns for better identification  
colnames(manu2\_conf)[c(2, 3)] <- paste0("conf\_", colnames(manu2\_conf)[c(2, 3)])  
colnames(manu2\_pred)[c(2, 3)] <- paste0("pred\_", colnames(manu2\_pred)[c(2, 3)])  
  
# Combine prediction and confidence intervals  
manu2\_pred\_combined <- cbind(manu2\_pred, manu2\_conf[, c(2, 3)])  
  
# Exponentiate predictions and intervals to back-transform from log space to original space  
manu2\_pred\_exp <- exp(manu2\_pred\_combined)  
  
# Exponentiate the log(manu) values to get the original 'manu' scale  
manu\_original <- exp(PD\_logmanu\_logSO2\_r167$manu\_LOG)  
  
# Order the X values for proper plotting  
X\_ordering <- order(manu\_original)  
X\_ordered <- manu\_original[X\_ordering]  
  
# Plot the data in original (non-logarithmic) space  
plot(  
 exp(SO2\_LOG) ~ manu\_original,  
 data = PD\_logmanu\_logSO2\_r167,  
 ylab = "SO2 (micrograms per cubic meter)",  
 xlab = "Number of large manufacturing plants (manu)",  
 main = "Conf and Pred of SO2 vs manu (exponentiated)",  
 pch = ".",   
 cex = 0,  
 xlim = c(min(manu\_original), max(manu\_original)),  
 ylim = c(min(exp(PD\_logmanu\_logSO2\_r167$SO2\_LOG)) - 0.4, max(exp(PD\_logmanu\_logSO2\_r167$SO2\_LOG)) + 0.4)  
)  
  
# Plot the exponentiated linear model (fitted values in original scale)  
lines(X\_ordered, manu2\_pred\_exp[X\_ordering, 1], col = "black", lwd = 1.7)  
  
# Plot the exponentiated confidence intervals  
matlines(  
 x = X\_ordered,  
 y = manu2\_pred\_exp[X\_ordering, c(4, 5)], # Columns for confidence interval bounds  
 col = "blue",  
 lty = 1,  
 lwd = 1.7  
)  
  
# Plot the exponentiated prediction intervals  
matlines(  
 x = X\_ordered,  
 y = manu2\_pred\_exp[X\_ordering, c(2, 3)], # Columns for prediction interval bounds  
 col = "red",  
 lty = 1,  
 lwd = 1.7  
)  
  
# Re-add the original points  
points(exp(SO2\_LOG) ~ manu\_original, data = PD\_logmanu\_logSO2\_r167, pch = 18L, cex = 1, col = 'darkorange')  
  
# Add a legend to distinguish between lines  
legend(  
 "topright",  
 legend = c("Linear model", "Conf. int.", "Pred. int."),  
 col = c("black", "blue", "red"),  
 lty = 1,  
 lwd = 1.8  
)



# Get confidence and prediction intervals for the linear model in log space  
predays2\_conf <- predict(predays2\_lm, interval = "conf")  
predays2\_pred <- predict(predays2\_lm, interval = "pred")

Warning in predict.lm(predays2\_lm, interval = "pred"): predictions on current data refer to \_future\_ responses

# Rename columns for better identification  
colnames(predays2\_conf)[c(2, 3)] <- paste0("conf\_", colnames(predays2\_conf)[c(2, 3)])  
colnames(predays2\_pred)[c(2, 3)] <- paste0("pred\_", colnames(predays2\_pred)[c(2, 3)])  
  
# Combine prediction and confidence intervals  
predays2\_pred\_combined <- cbind(predays2\_pred, predays2\_conf[, c(2, 3)])  
  
# Exponentiate predictions and intervals to back-transform SO2\_LOG from log space to original space  
predays2\_pred\_exp <- exp(predays2\_pred\_combined)  
  
# Order the X values for proper plotting  
X\_ordering <- order(PD\_predays\_logSO2\_r5$predays)  
X\_ordered <- PD\_predays\_logSO2\_r5$predays[X\_ordering]  
  
# Plot the data in original (non-logarithmic) space  
plot(  
 exp(SO2\_LOG) ~ predays,  
 data = PD\_predays\_logSO2\_r5,  
 ylab = "SO2 (micrograms per cubic meter)",  
 xlab = "Average annual number of precip days (predays)",  
 main = "Conf and Pred of SO2 vs predays (exponentiated)",  
 pch = ".",   
 cex = 0,  
 xlim = c(min(PD\_predays\_logSO2\_r5$predays), max(PD\_predays\_logSO2\_r5$predays)),  
 ylim = c(min(exp(PD\_predays\_logSO2\_r5$SO2\_LOG)) - 0.5, max(exp(PD\_predays\_logSO2\_r5$SO2\_LOG)) + 0.5)  
)  
  
# Plot the exponentiated linear model (fitted values in original scale)  
lines(X\_ordered, predays2\_pred\_exp[X\_ordering, 1], col = "black", lwd = 1.7)  
  
# Plot the exponentiated confidence intervals  
matlines(  
 x = X\_ordered,  
 y = predays2\_pred\_exp[X\_ordering, c(4, 5)], # Columns for confidence interval bounds  
 col = "blue",  
 lty = 1,  
 lwd = 1.7  
)  
  
# Plot the exponentiated prediction intervals  
matlines(  
 x = X\_ordered,  
 y = predays2\_pred\_exp[X\_ordering, c(2, 3)], # Columns for prediction interval bounds  
 col = "red",  
 lty = 1,  
 lwd = 1.7  
)  
  
# Re-add the original points in darkorange  
points(exp(SO2\_LOG) ~ predays, data = PD\_predays\_logSO2\_r5, pch = 18L, cex = 1, col = "deeppink1")  
  
# Add a legend to distinguish between lines  
legend(  
 "topright",  
 legend = c("Linear model", "Conf. int.", "Pred. int."),  
 col = c("black", "blue", "red"),  
 lty = 1,  
 lwd = 1.8  
)



## A4. Multiple Linear Regression

### A4.1 Fit Model using all Explanatory Variables

# Use Dummy variable data and omit 1 dummy when fitting  
PDD\_multi.lm <- lm(SO2 ~ manu + popul + precip + predays + temp + wind + region\_East + region\_North + region\_South, data=PD\_w\_dummies)  
summary(PDD\_multi.lm)

Call:  
lm(formula = SO2 ~ manu + popul + precip + predays + temp + wind +   
 region\_East + region\_North + region\_South, data = PD\_w\_dummies)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-23.034 -7.260 0.647 4.604 38.179   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 73.42613 46.43157 1.581 0.1239   
manu 0.06263 0.01404 4.461 9.99e-05 \*\*\*  
popul -0.03635 0.01355 -2.682 0.0116 \*   
precip 0.12899 0.36996 0.349 0.7297   
predays 0.07121 0.15108 0.471 0.6407   
temp -0.82201 0.62542 -1.314 0.1984   
wind -1.60524 1.68984 -0.950 0.3495   
region\_East 11.12061 8.17431 1.360 0.1835   
region\_North -9.83592 7.47949 -1.315 0.1981   
region\_South -4.90691 7.39307 -0.664 0.5118   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 13.02 on 31 degrees of freedom  
Multiple R-squared: 0.7614, Adjusted R-squared: 0.6922   
F-statistic: 10.99 on 9 and 31 DF, p-value: 1.869e-07

PDD\_multi\_rmanu.lm <- lm(SO2 ~ popul + precip + predays + temp + wind + region\_East + region\_North + region\_South, data=PD\_w\_dummies)  
summary(PDD\_multi\_rmanu.lm)

Call:  
lm(formula = SO2 ~ popul + precip + predays + temp + wind + region\_East +   
 region\_North + region\_South, data = PD\_w\_dummies)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-24.954 -10.807 -1.958 5.135 50.120   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 107.373923 57.769848 1.859 0.0723 .   
popul 0.021685 0.004786 4.531 7.73e-05 \*\*\*  
precip 0.278006 0.464701 0.598 0.5539   
predays 0.078287 0.190535 0.411 0.6839   
temp -1.584811 0.758762 -2.089 0.0448 \*   
wind -1.926758 2.129360 -0.905 0.3723   
region\_East 10.617952 10.308837 1.030 0.3107   
region\_North -11.574747 9.420658 -1.229 0.2282   
region\_South -5.958772 9.319737 -0.639 0.5271   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 16.42 on 32 degrees of freedom  
Multiple R-squared: 0.6083, Adjusted R-squared: 0.5104   
F-statistic: 6.212 on 8 and 32 DF, p-value: 7.519e-05

#transform So2 to be included in dummies   
PD\_w\_dummies$SO2\_LOG <- log(PD\_w\_dummies$SO2)  
  
PDD\_multi\_SO2LOG.lm <- lm(SO2\_LOG ~ manu + popul + precip + predays + temp + wind + region\_East + region\_North + region\_South, data=PD\_w\_dummies)  
summary(PDD\_multi\_SO2LOG.lm)

Call:  
lm(formula = SO2\_LOG ~ manu + popul + precip + predays + temp +   
 wind + region\_East + region\_North + region\_South, data = PD\_w\_dummies)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-0.84084 -0.21919 -0.04375 0.22586 0.84169   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 5.7780848 1.3439858 4.299 0.000158 \*\*\*  
manu 0.0011844 0.0004063 2.915 0.006553 \*\*   
popul -0.0006314 0.0003922 -1.610 0.117584   
precip 0.0068068 0.0107086 0.636 0.529677   
predays 0.0046781 0.0043730 1.070 0.292984   
temp -0.0417731 0.0181032 -2.308 0.027869 \*   
wind -0.1148447 0.0489132 -2.348 0.025437 \*   
region\_East 0.2769365 0.2366098 1.170 0.250745   
region\_North -0.3513192 0.2164978 -1.623 0.114773   
region\_South -0.3408820 0.2139963 -1.593 0.121322   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.3769 on 31 degrees of freedom  
Multiple R-squared: 0.7767, Adjusted R-squared: 0.7119   
F-statistic: 11.98 on 9 and 31 DF, p-value: 7.138e-08

### A4.2 Variable Selection

#### A4.2.1 Stepwise

step(PDD\_multi\_SO2LOG.lm, direction="backward", trace = 1)

Start: AIC=-71.47  
SO2\_LOG ~ manu + popul + precip + predays + temp + wind + region\_East +   
 region\_North + region\_South  
  
 Df Sum of Sq RSS AIC  
- precip 1 0.05741 4.4621 -72.936  
- predays 1 0.16260 4.5673 -71.981  
- region\_East 1 0.19465 4.5993 -71.694  
<none> 4.4047 -71.467  
- region\_South 1 0.36054 4.7652 -70.241  
- popul 1 0.36820 4.7729 -70.175  
- region\_North 1 0.37415 4.7788 -70.124  
- temp 1 0.75655 5.1612 -66.968  
- wind 1 0.78329 5.1880 -66.756  
- manu 1 1.20728 5.6120 -63.536  
  
Step: AIC=-72.94  
SO2\_LOG ~ manu + popul + predays + temp + wind + region\_East +   
 region\_North + region\_South  
  
 Df Sum of Sq RSS AIC  
<none> 4.4621 -72.936  
- region\_South 1 0.30819 4.7703 -72.198  
- region\_North 1 0.33161 4.7937 -71.997  
- region\_East 1 0.39302 4.8551 -71.475  
- popul 1 0.40004 4.8621 -71.416  
- wind 1 0.72752 5.1896 -68.743  
- predays 1 0.91528 5.3774 -67.286  
- temp 1 1.03407 5.4962 -66.390  
- manu 1 1.26560 5.7277 -64.699

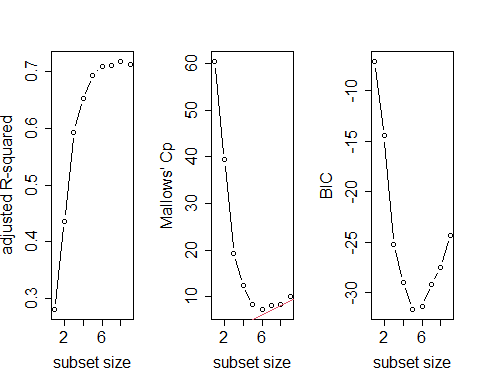
Call:  
lm(formula = SO2\_LOG ~ manu + popul + predays + temp + wind +   
 region\_East + region\_North + region\_South, data = PD\_w\_dummies)  
  
Coefficients:  
 (Intercept) manu popul predays temp   
 5.2116071 0.0012077 -0.0006551 0.0068627 -0.0333716   
 wind region\_East region\_North region\_South   
 -0.1078110 0.3475078 -0.3243140 -0.3020579

#### A4.2.2 All Subsets

#use dummy variables   
AllSubsets <- regsubsets(SO2\_LOG ~ manu + popul + precip + predays + temp + wind + region\_East + region\_North + region\_South, nvmax = 9, nbest = 1, data = PD\_w\_dummies)  
AllSubsets.summary <-summary(AllSubsets)  
AllSubsets.summary

Subset selection object  
Call: regsubsets.formula(SO2\_LOG ~ manu + popul + precip + predays +   
 temp + wind + region\_East + region\_North + region\_South,   
 nvmax = 9, nbest = 1, data = PD\_w\_dummies)  
9 Variables (and intercept)  
 Forced in Forced out  
manu FALSE FALSE  
popul FALSE FALSE  
precip FALSE FALSE  
predays FALSE FALSE  
temp FALSE FALSE  
wind FALSE FALSE  
region\_East FALSE FALSE  
region\_North FALSE FALSE  
region\_South FALSE FALSE  
1 subsets of each size up to 9  
Selection Algorithm: exhaustive  
 manu popul precip predays temp wind region\_East region\_North  
1 ( 1 ) " " " " " " " " "\*" " " " " " "   
2 ( 1 ) " " " " " " " " "\*" " " "\*" " "   
3 ( 1 ) "\*" " " " " " " "\*" " " "\*" " "   
4 ( 1 ) "\*" " " " " " " "\*" "\*" "\*" " "   
5 ( 1 ) "\*" " " " " "\*" "\*" "\*" "\*" " "   
6 ( 1 ) "\*" "\*" " " "\*" "\*" "\*" "\*" " "   
7 ( 1 ) "\*" "\*" "\*" " " "\*" "\*" " " "\*"   
8 ( 1 ) "\*" "\*" " " "\*" "\*" "\*" "\*" "\*"   
9 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
 region\_South  
1 ( 1 ) " "   
2 ( 1 ) " "   
3 ( 1 ) " "   
4 ( 1 ) " "   
5 ( 1 ) " "   
6 ( 1 ) " "   
7 ( 1 ) "\*"   
8 ( 1 ) "\*"   
9 ( 1 ) "\*"

# Modify some graphical parameters  
par(mfrow = c(1, 3))  
par(cex.axis = 1.5)  
par(cex.lab = 1.5)  
  
plot(1:9, AllSubsets.summary$adjr2, xlab = "subset size", ylab = "adjusted R-squared", type = "b")  
plot(1:9, AllSubsets.summary$cp, xlab = "subset size", ylab = "Mallows' Cp", type = "b")  
abline(0,1,col=2)  
plot(1:9, AllSubsets.summary$bic, xlab = "subset size", ylab = "BIC", type = "b")



#Comparing Brute Force 5 and Brute Force 6  
PDD\_multi\_BF5.lm <- lm(SO2\_LOG ~ manu + temp + predays + wind + region\_East, data = PD\_w\_dummies)  
summary(PDD\_multi\_BF5.lm)

Call:  
lm(formula = SO2\_LOG ~ manu + temp + predays + wind + region\_East,   
 data = PD\_w\_dummies)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-0.85758 -0.22315 -0.06617 0.13121 0.90555   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 5.6297637 0.9331379 6.033 6.98e-07 \*\*\*  
manu 0.0005519 0.0001133 4.871 2.37e-05 \*\*\*  
temp -0.0411032 0.0099925 -4.113 0.000225 \*\*\*  
predays 0.0062623 0.0025870 2.421 0.020818 \*   
wind -0.1365859 0.0470637 -2.902 0.006372 \*\*   
region\_East 0.6210108 0.1491331 4.164 0.000194 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.3889 on 35 degrees of freedom  
Multiple R-squared: 0.7317, Adjusted R-squared: 0.6934   
F-statistic: 19.09 on 5 and 35 DF, p-value: 3.94e-09

PDD\_multi\_BF6.lm <- lm(SO2\_LOG ~ manu + popul + temp + predays + wind + region\_East, data = PD\_w\_dummies)  
summary(PDD\_multi\_BF6.lm)

Call:  
lm(formula = SO2\_LOG ~ manu + popul + temp + predays + wind +   
 region\_East, data = PD\_w\_dummies)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-0.84832 -0.23689 -0.07347 0.23810 0.79084   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 5.3964520 0.9180635 5.878 1.24e-06 \*\*\*  
manu 0.0012245 0.0004059 3.017 0.00481 \*\*   
popul -0.0006707 0.0003896 -1.722 0.09421 .   
temp -0.0350622 0.0103371 -3.392 0.00178 \*\*   
predays 0.0056825 0.0025397 2.237 0.03192 \*   
wind -0.1298678 0.0459621 -2.826 0.00785 \*\*   
region\_East 0.6016508 0.1455517 4.134 0.00022 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.3784 on 34 degrees of freedom  
Multiple R-squared: 0.7532, Adjusted R-squared: 0.7097   
F-statistic: 17.3 on 6 and 34 DF, p-value: 4.695e-09

### A4.3 Making Models

#### A4.3.1 MLR1 Model Postulation - Pre-Transformation

summary(PDD\_multi\_BF5.lm)

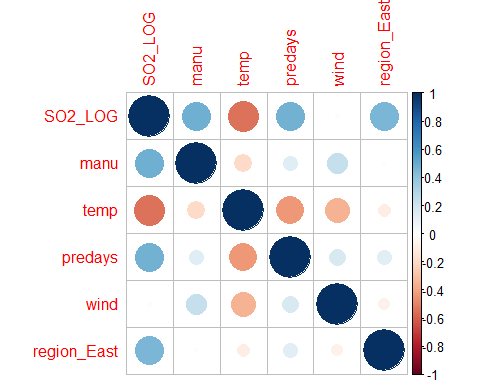
Call:  
lm(formula = SO2\_LOG ~ manu + temp + predays + wind + region\_East,   
 data = PD\_w\_dummies)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-0.85758 -0.22315 -0.06617 0.13121 0.90555   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 5.6297637 0.9331379 6.033 6.98e-07 \*\*\*  
manu 0.0005519 0.0001133 4.871 2.37e-05 \*\*\*  
temp -0.0411032 0.0099925 -4.113 0.000225 \*\*\*  
predays 0.0062623 0.0025870 2.421 0.020818 \*   
wind -0.1365859 0.0470637 -2.902 0.006372 \*\*   
region\_East 0.6210108 0.1491331 4.164 0.000194 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.3889 on 35 degrees of freedom  
Multiple R-squared: 0.7317, Adjusted R-squared: 0.6934   
F-statistic: 19.09 on 5 and 35 DF, p-value: 3.94e-09

#### A4.3.2 MLR1 Model Checking

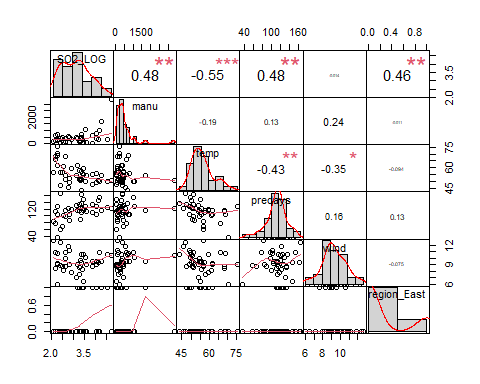
#Make model data set  
PD\_Multi5 <- data.frame(SO2\_LOG = PD\_w\_dummies$SO2\_LOG,   
 manu = PD\_w\_dummies$manu,   
 temp = PD\_w\_dummies$temp,   
 predays = PD\_w\_dummies$predays,   
 wind = PD\_w\_dummies$wind,   
 region\_East = PD\_w\_dummies$region\_East)  
  
# View the new DataFrame  
head(PD\_Multi5)

SO2\_LOG manu temp predays wind region\_East  
1 3.828641 44 47.6 135 8.8 1  
2 2.397895 46 56.8 58 8.9 0  
3 3.178054 368 61.5 115 9.1 1  
4 3.850148 625 55.0 111 9.6 1  
5 2.397895 391 47.1 166 12.4 0  
6 3.433987 35 55.2 148 6.5 0

#correlation matrix - large circles instead of just numbers  
corrplot(cor(PD\_Multi5))



suppressWarnings(chart.Correlation(PD\_Multi5))

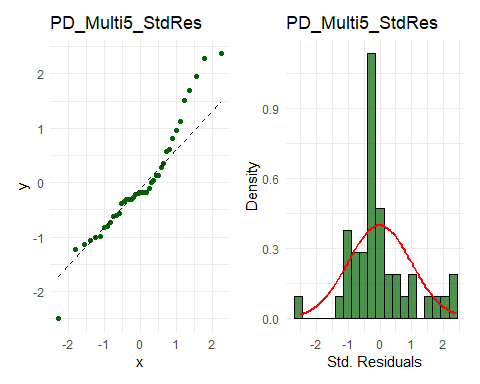


PD\_Multi5.lm <- lm(SO2\_LOG ~ manu + predays + temp + wind + region\_East, data=PD\_Multi5)  
  
vif(PD\_Multi5.lm)

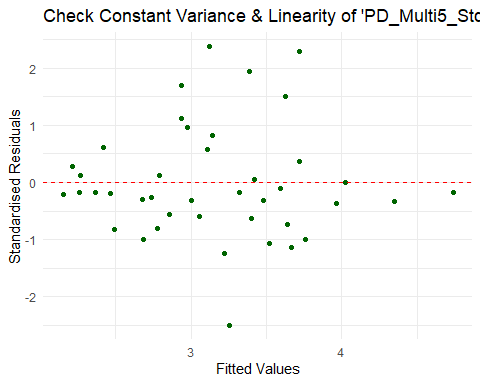
manu predays temp wind region\_East   
 1.078214 1.243719 1.379678 1.195778 1.033064

#### A4.3.3 MLR1 Checking linear regression assumptions with residuals

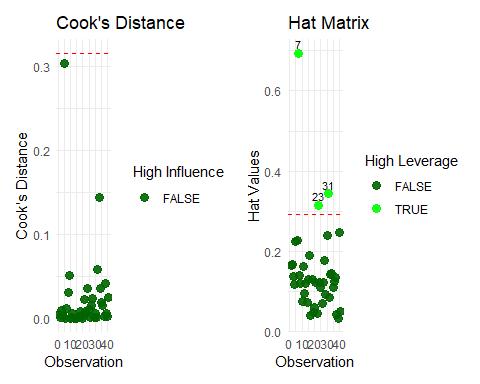
PD\_Multi5\_StdRes <- rstandard(PD\_Multi5.lm)  
  
PD\_Multi5\_StdRes\_qqplot <- ggplot\_qq(PD\_Multi5\_StdRes, "PD\_Multi5\_StdRes", "darkgreen")  
PD\_Multi5\_StdRes\_hist\_plot <- ggplot\_hist\_with\_normal(PD\_Multi5\_StdRes, "PD\_Multi5\_StdRes", "Std. Residuals", "darkgreen")  
  
PD\_Multi5\_resplot <- ggplot(PD\_Multi5, aes(x = PD\_Multi5.lm$fitted.values, y = PD\_Multi5\_StdRes)) +  
 geom\_point(color = 'darkgreen') +   
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +   
 labs(x = "Fitted Values", y = "Standardised Residuals",   
 title = "Check Constant Variance & Linearity of 'PD\_Multi5\_StdRes' Residuals") +  
 theme\_minimal()   
  
  
# Calculate the hat values  
PD\_Multi5\_HatValues <- hatvalues(PD\_Multi5.lm)  
 #Calculate p  
p <- length(coef(PD\_Multi5.lm)) - 1  
#calculate n  
n <- nrow(PD\_Multi5)  
# Combine hat values with the original data  
PD\_Multi5 <- PD\_Multi5 %>%   
 mutate(HatValues = PD\_Multi5\_HatValues,  
 HighLeverage = HatValues > (2 \* (p + 1) / n)) %>%  
 mutate(ID = row\_number())  
  
# Create the plot for Hat Values  
PD\_Multi5\_HatPlot <- ggplot(PD\_Multi5, aes(x = ID, y = HatValues)) +  
 geom\_point(aes(color = HighLeverage),   
 alpha = 0.9,   
 size = 3) +  
 scale\_color\_manual(values = c("FALSE" = "darkgreen", "TRUE" = "green"), name = "High Leverage") +  
 geom\_hline(yintercept = (2 \* (p + 1) / n), linetype = "dashed", color = "red") +   
 labs(x = "Observation", y = "Hat Values",   
 title = "Hat Matrix") +  
 theme\_minimal() +  
 geom\_text(data = filter(PD\_Multi5, HighLeverage),   
 aes(label = ID),   
 vjust = -0.5,   
 color = "black",   
 size = 3)  
  
  
  
PD\_Multi5\_Cooks <- cooks.distance(PD\_Multi5.lm)   
 #Calculate p  
p <- length(coef(PD\_Multi5.lm)) - 1  
#calculate n  
n <- nrow(PD\_Multi5)  
# Calculate Cook's distance and add to the dataset  
PD\_Multi5$CooksDistance <- PD\_Multi5\_Cooks  
PD\_Multi5 <- PD\_Multi5 %>%   
 mutate(HighInfluence = CooksDistance > (2 \* (p + 1) / (n - (p - 2)))) %>%  
 mutate(ID = row\_number())  
  
# Plot Cook's Distance  
PD\_Multi5\_CooksPlot <- ggplot(PD\_Multi5, aes(x = ID, y = CooksDistance)) +  
 geom\_point(aes(color = HighInfluence),   
 alpha = 0.9,   
 size = 3) +  
 scale\_color\_manual(values = c("FALSE" = "darkgreen", "TRUE" = "green"), name = "High Influence") +  
 geom\_hline(yintercept = (2 \* (p + 1) / (n - (p - 2))), linetype = "dashed", color = "red") +   
 labs(x = "Observation", y = "Cook's Distance",   
 title = "Cook's Distance") +  
 theme\_minimal() +  
 geom\_text(data = filter(PD\_Multi5, HighInfluence),   
 aes(label = ID),   
 vjust = -0.5,   
 color = "black",   
 size = 3)  
  
  
PD\_Multi5\_StdRes\_qqplot +   
PD\_Multi5\_StdRes\_hist\_plot



PD\_Multi5\_resplot



PD\_Multi5\_CooksPlot +   
PD\_Multi5\_HatPlot

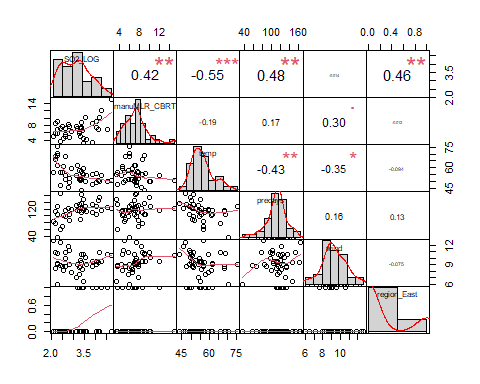


#### A4.3.4 MLR1 Transformation

#Transform manu  
manuMLR\_CBRT <- (PD\_w\_dummies$manu)^(1/3)  
  
#Make model data set  
PD\_Multi5\_manuCBRT <- data.frame(SO2\_LOG = PD\_w\_dummies$SO2\_LOG,   
 manuMLR\_CBRT = manuMLR\_CBRT,  
 temp = PD\_w\_dummies$temp,   
 predays = PD\_w\_dummies$predays,   
 wind = PD\_w\_dummies$wind,   
 region\_East = PD\_w\_dummies$region\_East)  
  
# View the new DataFrame  
head(PD\_Multi5\_manuCBRT)

SO2\_LOG manuMLR\_CBRT temp predays wind region\_East  
1 3.828641 3.530348 47.6 135 8.8 1  
2 2.397895 3.583048 56.8 58 8.9 0  
3 3.178054 7.166096 61.5 115 9.1 1  
4 3.850148 8.549880 55.0 111 9.6 1  
5 2.397895 7.312383 47.1 166 12.4 0  
6 3.433987 3.271066 55.2 148 6.5 0

#correlation matrix - large circles instead of just numbers  
#corrplot(cor(PD\_Multi5\_manuLOG))  
  
  
suppressWarnings(chart.Correlation(PD\_Multi5\_manuCBRT))



PD\_Multi5\_manuCBRT.lm <- lm(SO2\_LOG ~ manuMLR\_CBRT + predays + temp + wind + region\_East, data=PD\_Multi5\_manuCBRT)  
summary(PD\_Multi5\_manuCBRT.lm)

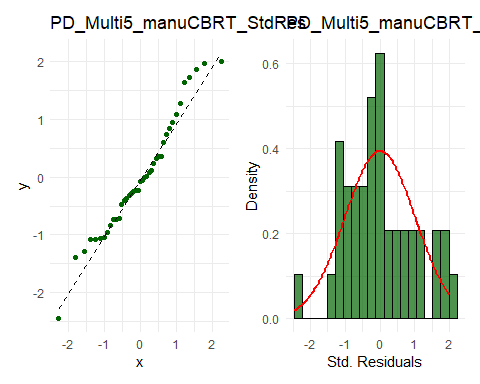
Call:  
lm(formula = SO2\_LOG ~ manuMLR\_CBRT + predays + temp + wind +   
 region\_East, data = PD\_Multi5\_manuCBRT)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-0.90649 -0.28823 -0.02964 0.21411 0.81084   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 5.293969 1.009970 5.242 7.72e-06 \*\*\*  
manuMLR\_CBRT 0.121119 0.030737 3.940 0.000371 \*\*\*  
predays 0.005886 0.002798 2.104 0.042655 \*   
temp -0.043331 0.010744 -4.033 0.000284 \*\*\*  
wind -0.146087 0.051563 -2.833 0.007598 \*\*   
region\_East 0.617913 0.160772 3.843 0.000490 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.4192 on 35 degrees of freedom  
Multiple R-squared: 0.6882, Adjusted R-squared: 0.6436   
F-statistic: 15.45 on 5 and 35 DF, p-value: 5.028e-08

vif(PD\_Multi5\_manuCBRT.lm)

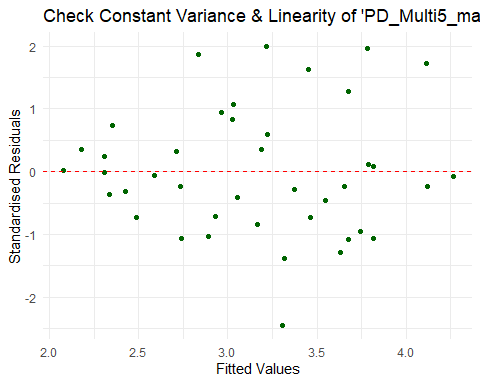
manuMLR\_CBRT predays temp wind region\_East   
 1.120537 1.251793 1.372337 1.234964 1.032992

#### A4.3.5 MLR1 Checking assumptions post\_transformation

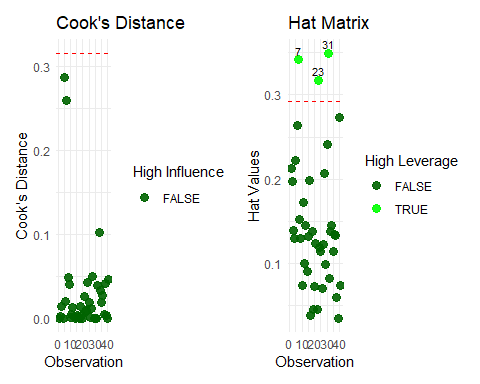
PD\_Multi5\_manuCBRT\_StdRes <- rstandard(PD\_Multi5\_manuCBRT.lm)  
  
PD\_Multi5\_manuCBRT\_StdRes\_qqplot <- ggplot\_qq(PD\_Multi5\_manuCBRT\_StdRes, "PD\_Multi5\_manuCBRT\_StdRes", "darkgreen")  
PD\_Multi5\_manuCBRT\_StdRes\_hist\_plot <- ggplot\_hist\_with\_normal(PD\_Multi5\_manuCBRT\_StdRes, "PD\_Multi5\_manuCBRT\_StdRes", "Std. Residuals", "darkgreen")  
  
PD\_Multi5\_manuCBRT\_resplot <- ggplot(PD\_Multi5\_manuCBRT, aes(x = PD\_Multi5\_manuCBRT.lm$fitted.values, y = PD\_Multi5\_manuCBRT\_StdRes)) +  
 geom\_point(color = 'darkgreen') +   
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +   
 labs(x = "Fitted Values", y = "Standardised Residuals",   
 title = "Check Constant Variance & Linearity of 'PD\_Multi5\_manuCBRT\_StdRes' Residuals") +  
 theme\_minimal()   
  
  
# Calculate the hat values  
PD\_Multi5\_manuCBRT\_HatValues <- hatvalues(PD\_Multi5\_manuCBRT.lm)  
 #Calculate p  
p <- length(coef(PD\_Multi5\_manuCBRT.lm)) - 1  
#calculate n  
n <- nobs(PD\_Multi5\_manuCBRT.lm)  
# Combine hat values with the original data  
PD\_Multi5\_manuCBRT <- PD\_Multi5\_manuCBRT %>%   
 mutate(HatValues = PD\_Multi5\_manuCBRT\_HatValues,  
 HighLeverage = HatValues > (2 \* (p + 1) / n)) %>%  
 mutate(ID = row\_number())  
  
# Create the plot for Hat Values  
PD\_Multi5\_manuCBRT\_HatPlot <- ggplot(PD\_Multi5\_manuCBRT, aes(x = ID, y = HatValues)) +  
 geom\_point(aes(color = HighLeverage),   
 alpha = 0.9,   
 size = 3) +  
 scale\_color\_manual(values = c("FALSE" = "darkgreen", "TRUE" = "green"), name = "High Leverage") +  
 geom\_hline(yintercept = (2 \* (p + 1) / n), linetype = "dashed", color = "red") +   
 labs(x = "Observation", y = "Hat Values",   
 title = "Hat Matrix") +  
 theme\_minimal() +  
 geom\_text(data = filter(PD\_Multi5\_manuCBRT, HighLeverage),   
 aes(label = ID),   
 vjust = -0.5,   
 color = "black",   
 size = 3)  
  
  
  
PD\_Multi5\_manuCBRT\_Cooks <- cooks.distance(PD\_Multi5\_manuCBRT.lm)   
 #Calculate p  
p <- length(coef(PD\_Multi5\_manuCBRT.lm)) - 1  
#calculate n  
n <- nobs(PD\_Multi5\_manuCBRT.lm)  
# Calculate Cook's distance and add to the dataset  
PD\_Multi5\_manuCBRT$CooksDistance <- PD\_Multi5\_manuCBRT\_Cooks  
PD\_Multi5\_manuCBRT <- PD\_Multi5\_manuCBRT %>%   
 mutate(HighInfluence = CooksDistance > (2 \* (p + 1) / (n - (p - 2)))) %>%  
 mutate(ID = row\_number())  
  
# Plot Cook's Distance  
PD\_Multi5\_manuCBRT\_CooksPlot <- ggplot(PD\_Multi5\_manuCBRT, aes(x = ID, y = CooksDistance)) +  
 geom\_point(aes(color = HighInfluence),   
 alpha = 0.9,   
 size = 3) +  
 scale\_color\_manual(values = c("FALSE" = "darkgreen", "TRUE" = "green"), name = "High Influence") +  
 geom\_hline(yintercept = (2 \* (p + 1) / (n - (p - 2))), linetype = "dashed", color = "red") +   
 labs(x = "Observation", y = "Cook's Distance",   
 title = "Cook's Distance") +  
 theme\_minimal() +  
 geom\_text(data = filter(PD\_Multi5\_manuCBRT, HighInfluence),   
 aes(label = ID),   
 vjust = -0.5,   
 color = "black",   
 size = 3)  
  
  
PD\_Multi5\_manuCBRT\_StdRes\_qqplot +   
PD\_Multi5\_manuCBRT\_StdRes\_hist\_plot



PD\_Multi5\_manuCBRT\_resplot



PD\_Multi5\_manuCBRT\_CooksPlot +   
PD\_Multi5\_manuCBRT\_HatPlot



#### A4.3.6 MLR1 Final Model 1 & ANOVA

summary(PD\_Multi5\_manuCBRT.lm)

Call:  
lm(formula = SO2\_LOG ~ manuMLR\_CBRT + predays + temp + wind +   
 region\_East, data = PD\_Multi5\_manuCBRT)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-0.90649 -0.28823 -0.02964 0.21411 0.81084   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 5.293969 1.009970 5.242 7.72e-06 \*\*\*  
manuMLR\_CBRT 0.121119 0.030737 3.940 0.000371 \*\*\*  
predays 0.005886 0.002798 2.104 0.042655 \*   
temp -0.043331 0.010744 -4.033 0.000284 \*\*\*  
wind -0.146087 0.051563 -2.833 0.007598 \*\*   
region\_East 0.617913 0.160772 3.843 0.000490 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.4192 on 35 degrees of freedom  
Multiple R-squared: 0.6882, Adjusted R-squared: 0.6436   
F-statistic: 15.45 on 5 and 35 DF, p-value: 5.028e-08

anova(PD\_Multi5\_manuCBRT.lm)

Analysis of Variance Table  
  
Response: SO2\_LOG  
 Df Sum Sq Mean Sq F value Pr(>F)   
manuMLR\_CBRT 1 3.4980 3.4980 19.901 8.07e-05 \*\*\*  
predays 1 3.3772 3.3772 19.214 0.0001016 \*\*\*  
temp 1 2.2133 2.2133 12.592 0.0011262 \*\*   
wind 1 1.8923 1.8923 10.766 0.0023466 \*\*   
region\_East 1 2.5964 2.5964 14.772 0.0004899 \*\*\*  
Residuals 35 6.1518 0.1758   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#### A4.4.1 MLR2 Model Postulation (alternative variable selection)

PDD\_multi\_T2 <- PD\_w\_dummies #make T2 dataframe  
  
PD\_w\_dummies$SO2\_LOG <- log(PD\_w\_dummies$SO2) # log SO2  
PDD\_multi\_T2$manu\_CBRT <- (PDD\_multi\_T2$manu)^(1/3) #cbrt manu  
PDD\_multi\_T2 <- PDD\_multi\_T2 %>% select( -ID, -region) #remove old  
  
  
PDD\_multi\_T2.lm <- lm(SO2\_LOG ~ manu\_CBRT + predays + temp + poly(wind, 3) + region\_East, data=PDD\_multi\_T2)  
summary(PDD\_multi\_T2.lm)

Call:  
lm(formula = SO2\_LOG ~ manu\_CBRT + predays + temp + poly(wind,   
 3) + region\_East, data = PDD\_multi\_T2)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-0.52267 -0.27141 -0.01544 0.24443 0.77334   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 4.090515 0.761506 5.372 6.16e-06 \*\*\*  
manu\_CBRT 0.088557 0.031146 2.843 0.007604 \*\*   
predays 0.006365 0.002647 2.404 0.021970 \*   
temp -0.043199 0.010016 -4.313 0.000137 \*\*\*  
poly(wind, 3)1 -1.198056 0.436559 -2.744 0.009731 \*\*   
poly(wind, 3)2 -0.753833 0.410147 -1.838 0.075086 .   
poly(wind, 3)3 -0.870353 0.418779 -2.078 0.045533 \*   
region\_East 0.572756 0.152472 3.756 0.000668 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.3907 on 33 degrees of freedom  
Multiple R-squared: 0.7447, Adjusted R-squared: 0.6905   
F-statistic: 13.75 on 7 and 33 DF, p-value: 3.541e-08

vif(PDD\_multi\_T2.lm)

GVIF Df GVIF^(1/(2\*Df))  
manu\_CBRT 1.324749 1 1.150977  
predays 1.290244 1 1.135889  
temp 1.373241 1 1.171854  
poly(wind, 3) 1.558943 3 1.076808  
region\_East 1.069801 1 1.034312

step(PDD\_multi\_T2.lm, direction="backward", trace = 1)

Start: AIC=-69.96  
SO2\_LOG ~ manu\_CBRT + predays + temp + poly(wind, 3) + region\_East  
  
 Df Sum of Sq RSS AIC  
<none> 5.0373 -69.965  
- predays 1 0.88249 5.9198 -65.346  
- manu\_CBRT 1 1.23407 6.2714 -62.981  
- poly(wind, 3) 3 2.52531 7.5626 -59.305  
- region\_East 1 2.15400 7.1913 -57.369  
- temp 1 2.83958 7.8769 -53.635

Call:  
lm(formula = SO2\_LOG ~ manu\_CBRT + predays + temp + poly(wind,   
 3) + region\_East, data = PDD\_multi\_T2)  
  
Coefficients:  
 (Intercept) manu\_CBRT predays temp poly(wind, 3)1   
 4.090515 0.088557 0.006365 -0.043199 -1.198056   
poly(wind, 3)2 poly(wind, 3)3 region\_East   
 -0.753833 -0.870353 0.572756

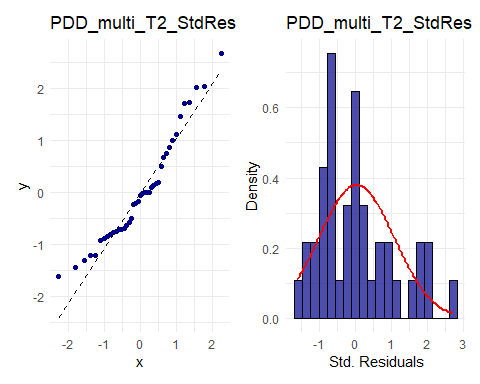
#### A4.4.2 Compare MLR2 Model to MLR1 Model - Partial F-test

anova(PD\_Multi5\_manuCBRT.lm, PDD\_multi\_T2.lm)

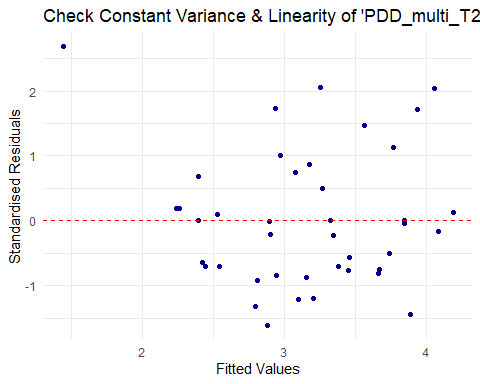
Analysis of Variance Table  
  
Model 1: SO2\_LOG ~ manuMLR\_CBRT + predays + temp + wind + region\_East  
Model 2: SO2\_LOG ~ manu\_CBRT + predays + temp + poly(wind, 3) + region\_East  
 Res.Df RSS Df Sum of Sq F Pr(>F)   
1 35 6.1518   
2 33 5.0373 2 1.1145 3.6505 0.03696 \*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#### A4.4.3 MLR2 Checking linear regression assumptions

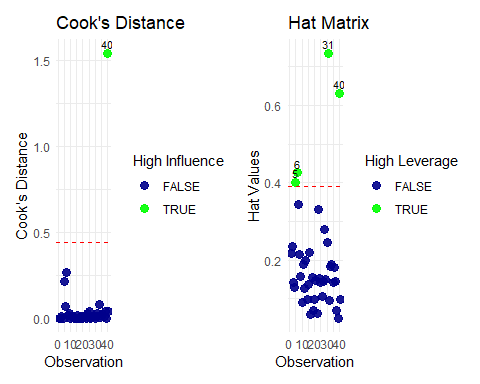
PDD\_multi\_T2\_StdRes <- rstandard(PDD\_multi\_T2.lm)  
  
PDD\_multi\_T2\_StdRes\_qqplot <- ggplot\_qq(PDD\_multi\_T2\_StdRes, "PDD\_multi\_T2\_StdRes", "darkblue")  
PDD\_multi\_T2\_StdRes\_hist\_plot <- ggplot\_hist\_with\_normal(PDD\_multi\_T2\_StdRes, "PDD\_multi\_T2\_StdRes", " Std. Residuals", "darkblue")  
  
PDD\_multi\_T2\_resplot <- ggplot(PDD\_multi\_T2, aes(x = PDD\_multi\_T2.lm$fitted.values, y = PDD\_multi\_T2\_StdRes)) +  
 geom\_point(color = 'darkblue') +   
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +   
 labs(x = "Fitted Values", y = "Standardised Residuals",   
 title = "Check Constant Variance & Linearity of 'PDD\_multi\_T2\_StdRes' Residuals") +  
 theme\_minimal()   
  
  
# Calculate the hat values  
PDD\_multi\_T2\_HatValues <- hatvalues(PDD\_multi\_T2.lm)  
 #Calculate p  
p <- length(coef(PDD\_multi\_T2.lm)) - 1  
#calculate n  
n <- nrow(PDD\_multi\_T2)  
# Combine hat values with the original data  
PDD\_multi\_T2 <- PDD\_multi\_T2 %>%   
 mutate(HatValues = PDD\_multi\_T2\_HatValues,  
 HighLeverage = HatValues > (2 \* (p + 1) / n)) %>%  
 mutate(ID = row\_number())  
  
# Create the plot for Hat Values  
PDD\_multi\_T2\_HatPlot <- ggplot(PDD\_multi\_T2, aes(x = ID, y = HatValues)) +  
 geom\_point(aes(color = HighLeverage),   
 alpha = 0.9,   
 size = 3) +  
 scale\_color\_manual(values = c("FALSE" = "darkblue", "TRUE" = "green"), name = "High Leverage") +  
 geom\_hline(yintercept = (2 \* (p + 1) / n), linetype = "dashed", color = "red") +   
 labs(x = "Observation", y = "Hat Values",   
 title = "Hat Matrix") +  
 theme\_minimal() +  
 geom\_text(data = filter(PDD\_multi\_T2, HighLeverage),   
 aes(label = ID),   
 vjust = -0.5,   
 color = "black",   
 size = 3)  
  
  
  
PDD\_multi\_T2\_Cooks <- cooks.distance(PDD\_multi\_T2.lm)   
# Calculate Cook's distance and add to the dataset  
PDD\_multi\_T2$CooksDistance <- PDD\_multi\_T2\_Cooks  
PDD\_multi\_T2 <- PDD\_multi\_T2 %>%   
 mutate(HighInfluence = CooksDistance > (2 \* (p + 1) / (n - (p - 2)))) %>%  
 mutate(ID = row\_number())  
  
# Plot Cook's Distance  
PDD\_multi\_T2\_CooksPlot <- ggplot(PDD\_multi\_T2, aes(x = ID, y = CooksDistance)) +  
 geom\_point(aes(color = HighInfluence),   
 alpha = 0.9,   
 size = 3) +  
 scale\_color\_manual(values = c("FALSE" = "darkblue", "TRUE" = "green"), name = "High Influence") +  
 geom\_hline(yintercept = (2 \* (p + 1) / (n - (p - 2))), linetype = "dashed", color = "red") +   
 labs(x = "Observation", y = "Cook's Distance",   
 title = "Cook's Distance") +  
 theme\_minimal() +  
 geom\_text(data = filter(PDD\_multi\_T2, HighInfluence),   
 aes(label = ID),   
 vjust = -0.5,   
 color = "black",   
 size = 3)  
  
  
PDD\_multi\_T2\_StdRes\_qqplot +   
PDD\_multi\_T2\_StdRes\_hist\_plot



PDD\_multi\_T2\_resplot



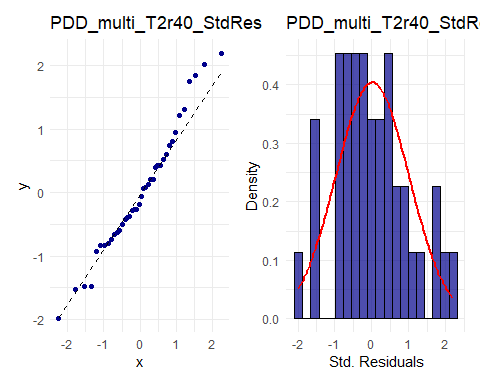
PDD\_multi\_T2\_CooksPlot +   
PDD\_multi\_T2\_HatPlot



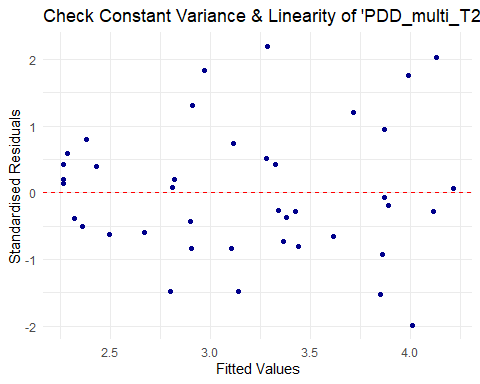
#### A4.4.4 Outlier Removal and reassess linear assumptions with residuals

PDD\_multi\_T2r40 <- PDD\_multi\_T2[-40, ]  
columns\_to\_remove <- c("ID", "HatValues", "HighLeverage", "CooksDistance", "HighInfluence")  
PDD\_multi\_T2r40 <- PDD\_multi\_T2r40[, !names(PDD\_multi\_T2r40) %in% columns\_to\_remove]  
# View(PDD\_multi\_T2r40)

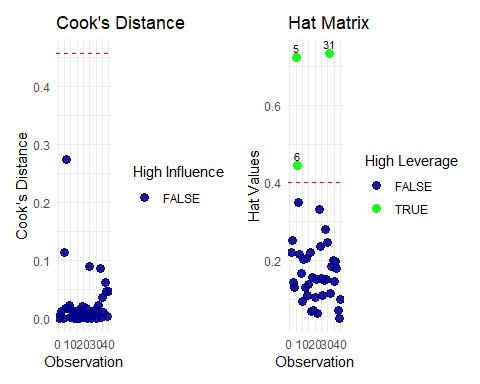
PDD\_multi\_T2r40.lm <- lm(SO2\_LOG ~ manu\_CBRT + predays + temp + poly(wind, 3) + region\_East, data=PDD\_multi\_T2r40)  
  
  
  
PDD\_multi\_T2r40\_StdRes <- rstandard(PDD\_multi\_T2r40.lm)  
  
PDD\_multi\_T2r40\_StdRes\_qqplot <- ggplot\_qq(PDD\_multi\_T2r40\_StdRes, "PDD\_multi\_T2r40\_StdRes", "darkblue")  
PDD\_multi\_T2r40\_StdRes\_hist\_plot <- ggplot\_hist\_with\_normal(PDD\_multi\_T2r40\_StdRes, "PDD\_multi\_T2r40\_StdRes", " Std. Residuals", "darkblue")  
  
PDD\_multi\_T2r40\_resplot <- ggplot(PDD\_multi\_T2r40, aes(x = PDD\_multi\_T2r40.lm$fitted.values, y = PDD\_multi\_T2r40\_StdRes)) +  
 geom\_point(color = 'darkblue') +   
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +   
 labs(x = "Fitted Values", y = "Standardised Residuals",   
 title = "Check Constant Variance & Linearity of 'PDD\_multi\_T2r40\_StdRes' Residuals") +  
 theme\_minimal()   
  
  
# Calculate the hat values  
PDD\_multi\_T2r40\_HatValues <- hatvalues(PDD\_multi\_T2r40.lm)  
 #Calculate p  
p <- length(coef(PDD\_multi\_T2r40.lm)) - 1  
#calculate n  
n <- nrow(PDD\_multi\_T2r40)  
# Combine hat values with the original data  
PDD\_multi\_T2r40 <- PDD\_multi\_T2r40 %>%   
 mutate(HatValues = PDD\_multi\_T2r40\_HatValues,  
 HighLeverage = HatValues > (2 \* (p + 1) / n)) %>%  
 mutate(ID = row\_number())  
  
# Create the plot for Hat Values  
PDD\_multi\_T2r40\_HatPlot <- ggplot(PDD\_multi\_T2r40, aes(x = ID, y = HatValues)) +  
 geom\_point(aes(color = HighLeverage),   
 alpha = 0.9,   
 size = 3) +  
 scale\_color\_manual(values = c("FALSE" = "darkblue", "TRUE" = "green"), name = "High Leverage") +  
 geom\_hline(yintercept = (2 \* (p + 1) / n), linetype = "dashed", color = "red") +   
 labs(x = "Observation", y = "Hat Values",   
 title = "Hat Matrix") +  
 theme\_minimal() +  
 geom\_text(data = filter(PDD\_multi\_T2r40, HighLeverage),   
 aes(label = ID),   
 vjust = -0.5,   
 color = "black",   
 size = 3)  
  
  
PDD\_multi\_T2r40\_Cooks <- cooks.distance(PDD\_multi\_T2r40.lm)   
# Calculate Cook's distance and add to the dataset  
PDD\_multi\_T2r40$CooksDistance <- PDD\_multi\_T2r40\_Cooks  
PDD\_multi\_T2r40 <- PDD\_multi\_T2r40 %>%   
 mutate(HighInfluence = CooksDistance > (2 \* (p + 1) / (n - (p - 2)))) %>%  
 mutate(ID = row\_number())  
  
# Plot Cook's Distance  
PDD\_multi\_T2r40\_CooksPlot <- ggplot(PDD\_multi\_T2r40, aes(x = ID, y = CooksDistance)) +  
 geom\_point(aes(color = HighInfluence),   
 alpha = 0.9,   
 size = 3) +  
 scale\_color\_manual(values = c("FALSE" = "darkblue", "TRUE" = "green"), name = "High Influence") +  
 geom\_hline(yintercept = (2 \* (p + 1) / (n - (p - 2))), linetype = "dashed", color = "red") +   
 labs(x = "Observation", y = "Cook's Distance",   
 title = "Cook's Distance") +  
 theme\_minimal() +  
 geom\_text(data = filter(PDD\_multi\_T2r40, HighInfluence),   
 aes(label = ID),   
 vjust = -0.5,   
 color = "black",   
 size = 3)  
  
  
PDD\_multi\_T2r40\_StdRes\_qqplot +   
PDD\_multi\_T2r40\_StdRes\_hist\_plot



PDD\_multi\_T2r40\_resplot



PDD\_multi\_T2r40\_CooksPlot +   
PDD\_multi\_T2r40\_HatPlot



#### A4.4.5 MLR2 Final Model 2 & ANOVA

summary(PDD\_multi\_T2r40.lm)

Call:  
lm(formula = SO2\_LOG ~ manu\_CBRT + predays + temp + poly(wind,   
 3) + region\_East, data = PDD\_multi\_T2r40)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-0.64160 -0.19807 -0.04027 0.13768 0.74142   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 3.941937 0.687321 5.735 2.34e-06 \*\*\*  
manu\_CBRT 0.085605 0.027979 3.060 0.004458 \*\*   
predays 0.008926 0.002526 3.533 0.001273 \*\*   
temp -0.045056 0.009013 -4.999 1.99e-05 \*\*\*  
poly(wind, 3)1 -1.204763 0.407550 -2.956 0.005809 \*\*   
poly(wind, 3)2 -0.830146 0.359727 -2.308 0.027634 \*   
poly(wind, 3)3 -1.412653 0.387427 -3.646 0.000935 \*\*\*  
region\_East 0.539668 0.137330 3.930 0.000427 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.3508 on 32 degrees of freedom  
Multiple R-squared: 0.7877, Adjusted R-squared: 0.7413   
F-statistic: 16.97 on 7 and 32 DF, p-value: 3.88e-09

anova(PDD\_multi\_T2r40.lm)

Analysis of Variance Table  
  
Response: SO2\_LOG  
 Df Sum Sq Mean Sq F value Pr(>F)   
manu\_CBRT 1 3.0085 3.00851 24.454 2.331e-05 \*\*\*  
predays 1 2.9572 2.95723 24.037 2.636e-05 \*\*\*  
temp 1 2.3716 2.37159 19.277 0.0001156 \*\*\*  
poly(wind, 3) 3 4.3734 1.45781 11.849 2.212e-05 \*\*\*  
region\_East 1 1.8999 1.89990 15.443 0.0004265 \*\*\*  
Residuals 32 3.9369 0.12303   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

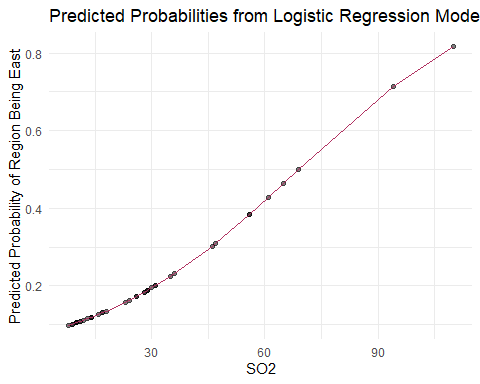
## A5 Extension

ExtData <- PD\_w\_dummies  
  
#SO2 as x var, East as y var  
model <- glm(region\_East ~ SO2, data = ExtData, family = binomial)  
summary(model)

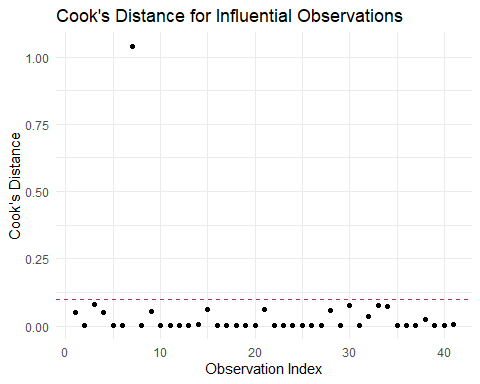
Call:  
glm(formula = region\_East ~ SO2, family = binomial, data = ExtData)  
  
Coefficients:  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) -2.51267 0.73730 -3.408 0.000655 \*\*\*  
SO2 0.03645 0.01689 2.158 0.030930 \*   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
(Dispersion parameter for binomial family taken to be 1)  
  
 Null deviance: 43.156 on 40 degrees of freedom  
Residual deviance: 37.677 on 39 degrees of freedom  
AIC: 41.677  
  
Number of Fisher Scoring iterations: 4

confint\_model <- confint(model)

# Predicted probabilities  
predicted\_probs <- predict(model, type = "response")  
  
# Add predicted probabilities to ExtData  
ExtData$predicted\_prob <- predicted\_probs  
  
# Plot predicted probabilities vs. SO2  
ggplot(ExtData, aes(x = SO2, y = predicted\_prob)) +  
 geom\_point(alpha = 0.5) +  
 geom\_line(aes(y = predicted\_prob), color = "maroon") +  
 labs(x = "SO2", y = "Predicted Probability of Region Being East") +  
 theme\_minimal() +  
 ggtitle("Predicted Probabilities from Logistic Regression Model")



# Calculate Cook's distance  
cooks\_distance <- cooks.distance(model)  
  
# Create a plot of Cook's distance  
ggplot(data = data.frame(Index = 1:length(cooks\_distance), CookD = cooks\_distance),   
 aes(x = Index, y = CookD)) +  
 geom\_point() +  
 geom\_hline(yintercept = 4 / length(cooks\_distance), linetype = "dashed", color = "maroon") +  
 labs(x = "Observation Index", y = "Cook's Distance") +  
 theme\_minimal() +  
 ggtitle("Cook's Distance for Influential Observations")



#new dataframe  
SO2\_values <- seq(min(ExtData$SO2), max(ExtData$SO2), length.out = 100)  
predicted\_values <- data.frame(SO2 = SO2\_values)  
  
# Calculate predicted probabilities for new SO2 values  
predicted\_probs\_new <- predict(model, newdata = predicted\_values, type = "response")  
  
# Calculate standard errors for predicted probabilities  
se\_fit <- sqrt(predicted\_probs\_new \* (1 - predicted\_probs\_new) / nrow(ExtData))  
  
# Calculate lower and upper confidence intervals  
lower\_ci <- predicted\_probs\_new - 1.96 \* se\_fit  
upper\_ci <- predicted\_probs\_new + 1.96 \* se\_fit  
  
# Create a new data frame for plotting with confidence intervals  
plot\_data <- data.frame(SO2 = SO2\_values,  
 predicted\_prob = predicted\_probs\_new,  
 lower\_ci = lower\_ci,  
 upper\_ci = upper\_ci)  
  
# Plot predicted probabilities with confidence intervals as a ribbon  
ggplot(plot\_data, aes(x = SO2)) +  
 geom\_line(aes(y = predicted\_prob), color = "maroon") +  
 geom\_ribbon(aes(ymin = lower\_ci, ymax = upper\_ci), alpha = 0.2, fill = "lightblue") +  
 labs(x = "SO2", y = "Predicted Probability of Region Being East") +  
 theme\_minimal() +  
 ggtitle("Predicted Probabilities with Confidence Intervals with Logistic GLM")

