Final Project-Final Edition

Fayed Patel

12/8/2019

Overview

The data on pollution in Singapore is comprised of multiple things.

Breakdown

(variables that need explaining)

- CO
  - Known as Carbon Monoxide. This is a gas that's emitted by anything that runs on burning fuel. such as a Gas Car.
- NO2
  - Known as Nitrogen Dioxide. This is also caused by buring fossil fuels.
- O3
  - Known as the Ozone. The Ozone here is referring to ground level Ozone which can be toxic.
- SO2
  - Known as Sulphur Dioxide. Alot of this is emmited by Coal and oil. some emmited by metal smelting and other manufacturing.
- pm
  - Known as particulate matter. This means the amount of solid and liquid in the air. Usually you would hear about pm10 or pm2.5. The numbers after pm mean micrometers of particulate matter. EX: Human is is usually 100 micrometers in diameter. pm10 and pm2.5 basically things like dust, smoke, etc

"What am I doing" you may ask?

Well I'm using the pollution data and showing that there is an increase of pollution due to the increase vehicles, commerical buildings, residentual buildings, and industry types.

The plan is to seperate the data as good as possible so that I can show this increase by specific variables then to plaster them on to a graph (mostly line graphs) to show that the trends are the same.

Hypothesis

The hypothesis is the increase of pollution due to energy consumption being increased by how much is produced to support people in their daily lives.

The Plan (Early Stages)

Reading in Data

We're going to first import the library Tidyverse for a bit of dplyr and ggplot2. Well todo any modeling first I'm going to need to read in my data. Since theres soo many CSV files, I thought it would take forever just manually read them in . So after some research I found a way to automatically read in all the CSV files in the folder without doing it individually.

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

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

```
#library(tidyverse)

setwd("/home/lucious/Documents/Final Proj")
temp = list.files(pattern = ".csv")
list2env(
  lapply(setNames(temp,
                  make.names(gsub("*.csv$", "", temp))),
         read.csv), envir = .GlobalEnv)
```

Now we check for any Nas

```
sum(is.na(CO))
sum(is.na(Commercial_buildings))
sum(is.na(Industry_values))
sum(is.na(NO2))
sum(is.na(O3))
sum(is.na(Pb))
sum(is.na(pm10))
sum(is.na(pm2.5))
sum(is.na(Residential_building))
sum(is.na(SO2))
sum(is.na(vehicle.population))
```

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

No Nas at all in any of my CSV files. I can assume that my data is clean to begin with and this lets me start doing an overview of the data i'll be working with.

Merging the data

```
mergedGases <- Reduce(function(x,y) merge(x,y, by="year")
                      ,list(CO, NO2, O3, SO2))
mergedGases

summary(mergedGases)
sd(mergedGases$CO_mean)
sd(mergedGases$NO2_mean)
sd(mergedGases$O3_mean)
sd(mergedGases$SO2_mean)
```

```
##    year CO_mean NO2_mean O3_mean SO2_mean
## 1  1999     3.6       36     125       22
## 2  2000     3.7       30     108       22
## 3  2001     4.2       26     126       22
## 4  2002     2.8       27     114       18
## 5  2003     3.1       24     108       15
## 6  2004     2.8       26     143       14
## 7  2005     2.4       25     155       14
## 8  2006     2.6       24     127       11
## 9  2007     1.7       22     140       12
## 10 2008     1.5       22     103       11
## 11 2009     1.7       22     100        9
## 12 2010     2.2       23     129       11
## 13 2011     2.0       25     110       10
## 14 2012     1.9       25     122       13
## 15 2013     5.5       25     139       14
## 16 2014     1.8       24     135       12
##       year         CO_mean         NO2_mean        O3_mean
##  Min.   :1999   Min.   :1.500   Min.   :22.00   Min.   :100.0
##  1st Qu.:2003   1st Qu.:1.875   1st Qu.:23.75   1st Qu.:109.5
##  Median :2006   Median :2.500   Median :25.00   Median :125.5
##  Mean   :2006   Mean   :2.719   Mean   :25.38   Mean   :124.0
##  3rd Qu.:2010   3rd Qu.:3.225   3rd Qu.:26.00   3rd Qu.:136.0
##  Max.   :2014   Max.   :5.500   Max.   :36.00   Max.   :155.0
##     SO2_mean
##  Min.   : 9.00
##  1st Qu.:11.00
##  Median :13.50
##  Mean   :14.38
##  3rd Qu.:15.75
##  Max.   :22.00
## [1] 1.089476
## [1] 3.5
## [1] 15.89969
## [1] 4.349329
```

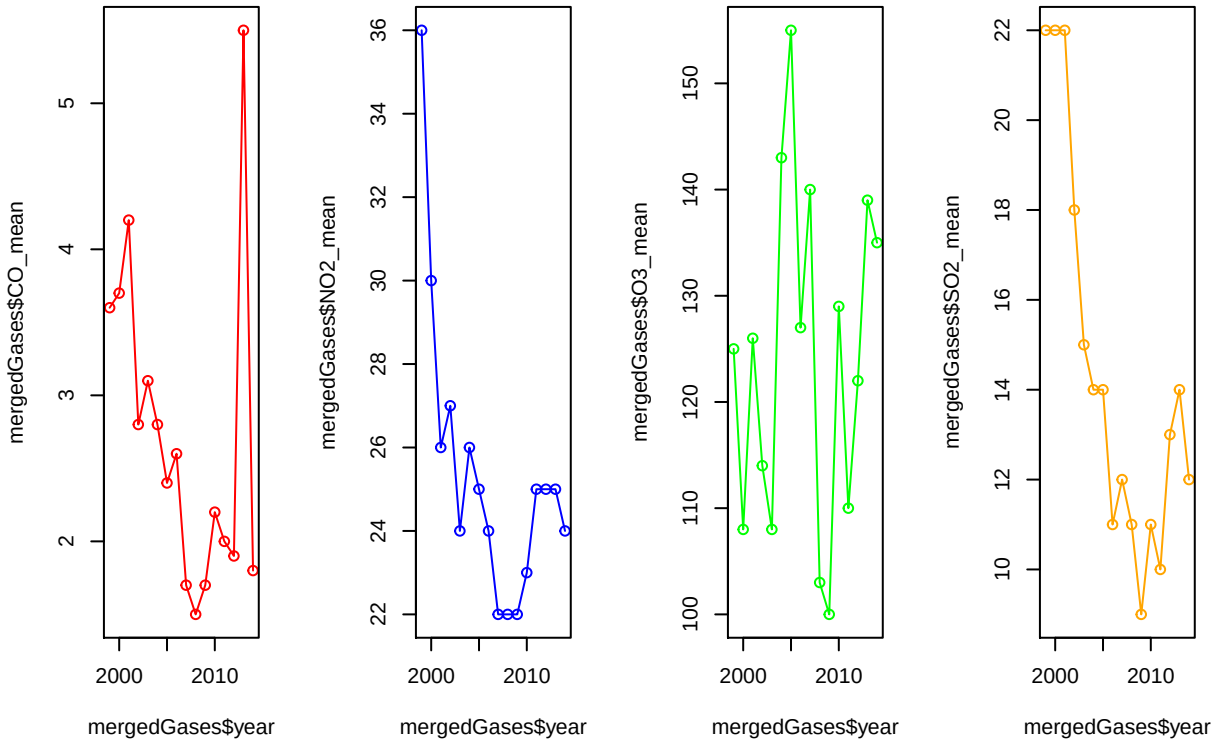
Since some of the files were seperate, i decided to combine them all into 1 dataframe by year. all 4 of these were from 1999 - 2014 conveniently.

```r
gaspm <- Reduce(function(x,y) merge(x,y, by ="year"),
                list(mergedGases, pm10, pm2.5))
#gaspm

summary(gaspm$pm10_mean)
summary(gaspm$pm2.5_mean)
sd(gaspm$pm10_mean)
sd(gaspm$pm2.5_mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    55.0    69.0    83.0   105.5   127.0   228.0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.00   18.00   19.00   19.38   21.00   23.00
## [1] 57.78353
## [1] 2.18092
```

Extra plots

```r
par(mfrow = c(1,4))
plot(mergedGases$year, mergedGases$CO_mean,
     type = "o", col = "red")
plot(mergedGases$year, mergedGases$NO2_mean,
     type = "o", col = "blue")
plot(mergedGases$year, mergedGases$O3_mean,
     type = "o", col = "green")
plot(mergedGases$year, mergedGases$SO2_mean,
     type = "o", col = "Orange")
```




```
par(mfrow = c(1,1))
```

Just wanted to see what plotting it would look like.

Data seperation

I merged the data of gases with pm. this decreased the number of observations 16 to 13. I can only assume this happened because the advancement in technology for Singapore and so as a result got more data. (will do a bit more research on why there are only records of pm starting at 2002)

```
sepindustry <- split(Industry_values, Industry_values$product_type)
#sepindustry[1]

sepcombuild <- split(Commercial_buildings, Commercial_buildings$building_type)
#sepcombuild[1]

sepresbuild <- split(Residential_building, Residential_building$building_type)
#sepresbuild[1]

sepvehiclepop <- split(vehicle.population, vehicle.population$car_type)
#sepvehiclepop[1]
```

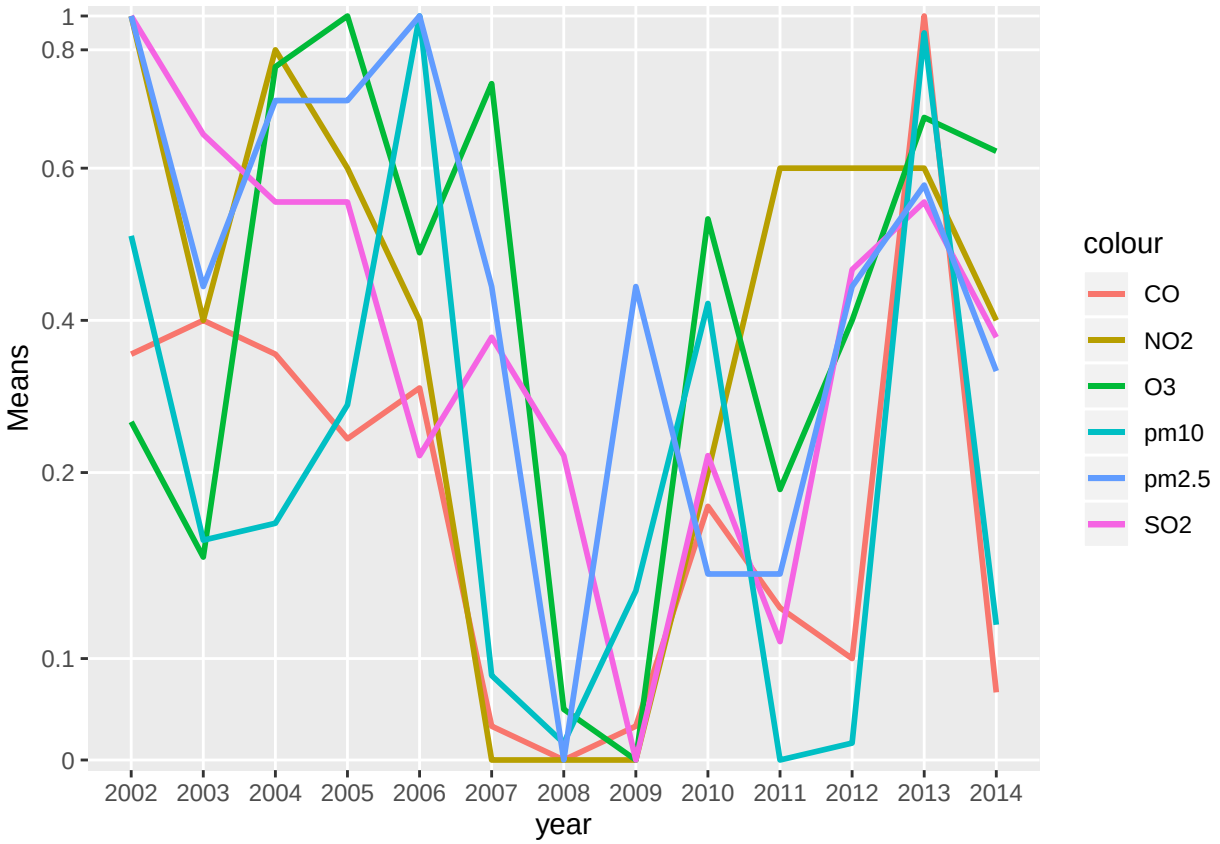
So here I wanted to split the data by type to for later so that I could see which type contributes more to the pollution. As an extra, if I find the time, I want to show if a reduction in a certain type would help allivate the pollution more than the others.

Scaling and Plotting

```
scaledpoll <- as.data.frame(apply(gaspm[,-c(1)], 2, function(x) (x - min(x))/(max(x)-min(x))))
scaledpoll$year <- as.factor(gaspm$year)
scaledpoll$CO_mean <- as.factor(scaledpoll$CO_mean)
scaledpoll$NO2_mean <- as.factor(scaledpoll$NO2_mean)
scaledpoll$O3_mean <- as.factor(scaledpoll$O3_mean)
scaledpoll$SO2_mean <- as.factor(scaledpoll$SO2_mean)
scaledpoll$pm10_mean <- as.factor(scaledpoll$pm10_mean)
scaledpoll$pm2.5_mean <- as.factor(scaledpoll$pm2.5_mean)
```

So here I scaled my data due to the vaulues differing. Some columns being "0.2" and others being "100"+. By scaling the data I'm able to plot them without any problems of some not showing up since the plots for some variables are smaller than others. By using the `as.factor()` I'll be using this for ggplot.

```
ggplot(scaledpoll, aes(x = year, group = 0)) +
#  theme_bw() +
  geom_line(aes(y = CO_mean, colour = "CO"), size = 1) +
  geom_line(aes(y = NO2_mean, colour = "NO2"), size = 1) +
  geom_line(aes(y = O3_mean, colour = "O3"), size = 1) +
  geom_line(aes(y = SO2_mean, colour = "SO2"), size = 1) +
  geom_line(aes(y = pm10_mean, colour = "pm10"), size = 1) +
  geom_line(aes(y = pm2.5_mean, colour = "pm2.5"), size = 1) +
  scale_y_discrete(breaks = seq(from=0, to=1, by=0.1)) +
  ylab("Means")
```



I plot the amount of pollutants by year just to get view on the change in output of the pollutants through

each year.

The Plan (Mid Stage)

Prepping the dataset

```
indusvalue <- data.frame(sepindustry$`Basic Metal`$year)
indusvalue$`Basic Metal` <- sepindustry$`Basic Metal`$values
indusvalue$`Chemicals & Chemical Products` <- sepindustry$`Chemicals & Chemical Products`$values
indusvalue$`Computer, Electronic & Optical Products` <- sepindustry$`Computer, Electronic & Optical Proc
indusvalue$`Electrical Equipment` <- sepindustry$`Electrical Equipment`$values
indusvalue$`Fabricated Metal Products` <- sepindustry$`Fabricated Metal Products`$values
indusvalue$`Food, Beverage & Tobacco` <- sepindustry$`Food, Beverage & Tobacco`$values
indusvalue$Furniture <- sepindustry$Furniture$values
indusvalue$`Leather, Leather Products & Footwear` <- sepindustry$`Leather, Leather Products & Footwear`$
indusvalue$`Machinery & Equipment` <- sepindustry$`Machinery & Equipment`$values
indusvalue$`Motor Vehicles, Trailers & Semi-trailers` <- sepindustry$`Motor Vehicles, Trailers & Semi-t
indusvalue$`Non-metallic Mineral Products` <- sepindustry$`Non-metallic Mineral Products`$values
indusvalue$`Other Manufacturing Industries` <- sepindustry$`Other Manufacturing Industries`$values
indusvalue$`Other Transport Equipment` <- sepindustry$`Other Transport Equipment`$values
indusvalue$`Paper & Paper Products` <- sepindustry$`Paper & Paper Products`$values
indusvalue$`Pharmaceutical & Biological Products` <- sepindustry$`Pharmaceutical & Biological Products`$
indusvalue$`Printing & Reproduction Of Recorded Media` <- sepindustry$`Printing & Reproduction Of Record
indusvalue$`Refined Petroleum Products` <- sepindustry$`Refined Petroleum Products`$values
indusvalue$`Rubber & Plastic Products` <- sepindustry$`Rubber & Plastic Products`$values
indusvalue$Textiles <- sepindustry$Textiles$values
indusvalue$`Wearing Apparel` <- sepindustry$`Wearing Apparel`$values
indusvalue$`Wood & Wood Products` <- sepindustry$`Wood & Wood Products`$values
colnames(indusvalue)[colnames(indusvalue) == "sepindustry..Basic.Metal..year"] <- "year"
```

Couldn't figure how to automate it, so I manually made the the dataset. This dataset contains the the Output Produced by year. the dataset was all the same dimensions, so no problems occured.

Cor Testing

```
induspol <- merge(gaspm, indusvalue, by = "year")

corlist <- cor(induspol)
corlist2 <- data.frame(corlist[2:7, 8:28])

corlist3 <- corlist2 %>%
  #filter_all(any_vars(abs(.) > 0.67))
  select_if(funs(any(abs(.) > 0.6)))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
```

```
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.
```

```
corlist3
```

```
##            Basic.Metal Electrical.Equipment Fabricated.Metal.Products
## CO_mean    -0.45976799          -0.35407082               -0.20430856
## NO2_mean   -0.57649515          -0.66885038               -0.50915077
## O3_mean    -0.01884408           0.08942438               -0.04486569
## SO2_mean   -0.69966441          -0.63136977               -0.71529012
## pm10_mean  -0.22224482          -0.05798085               -0.09827625
## pm2.5_mean -0.58412926          -0.33277071               -0.60711892
##            Other.Transport.Equipment Paper...Paper.Products
## CO_mean                  -0.12832572            -0.14869171
## NO2_mean                 -0.47436369            -0.48492900
## O3_mean                  -0.06813445             0.13588765
## SO2_mean                 -0.67061945            -0.64655899
## pm10_mean                -0.03795986            -0.06475424
## pm2.5_mean               -0.56178250            -0.66236546
##            Pharmaceutical...Biological.Products Refined.Petroleum.Products
## CO_mean                              -0.3087615                -0.19454844
## NO2_mean                             -0.3273766                -0.39146871
## O3_mean                               0.1289230                 0.03215212
## SO2_mean                             -0.6737095                -0.61255871
## pm10_mean                            -0.1234104                -0.12023436
## pm2.5_mean                           -0.3172155                -0.54548612
##            Rubber...Plastic.Products   Textiles
## CO_mean                  -0.00421154 0.08618810
## NO2_mean                  0.24190577 0.31469697
## O3_mean                   0.14845220 0.07772376
## SO2_mean                  0.48644266 0.60140702
## pm10_mean                 0.21119155 0.05558238
## pm2.5_mean                0.60095017 0.56360429
```

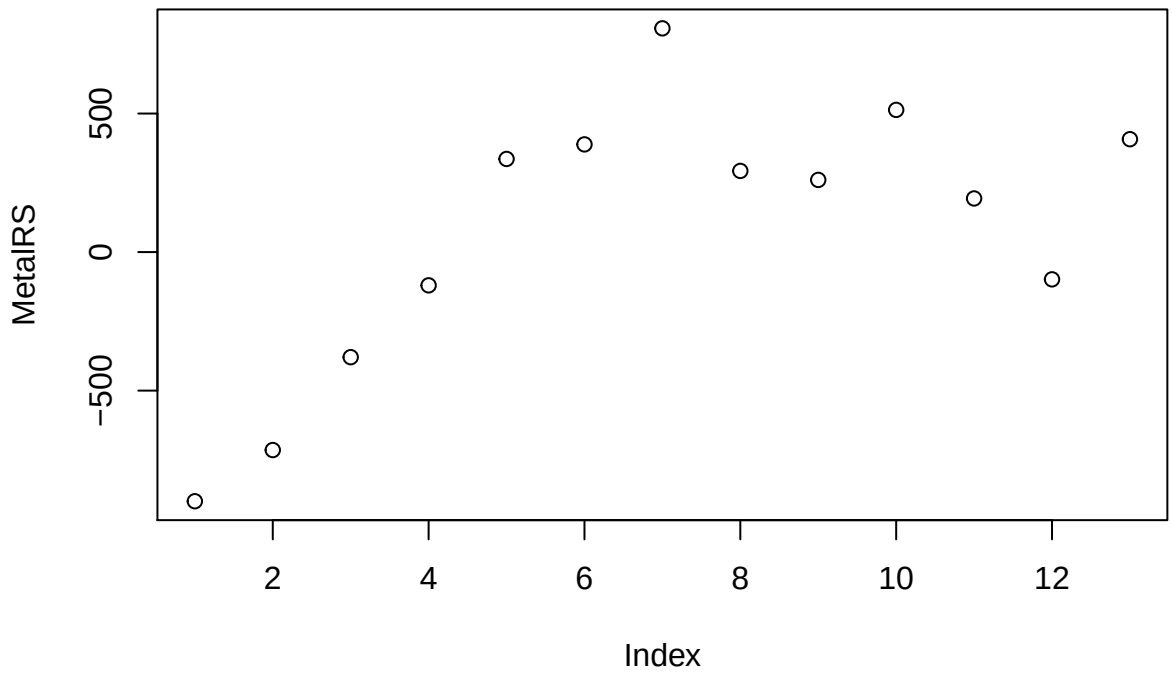
I mergeed the pollutants and the dataset made above by year to keep the dimensions consistant but also to add the pollutants that we need to run the `cor()` function on our industries. Here we get glimpse of dplyr and I use it to run a cor test on my entire dataset without doing it manually and adding the condition that if there's any correlation that is higher than 0.6 regardless if its negative or Positive.

Regression

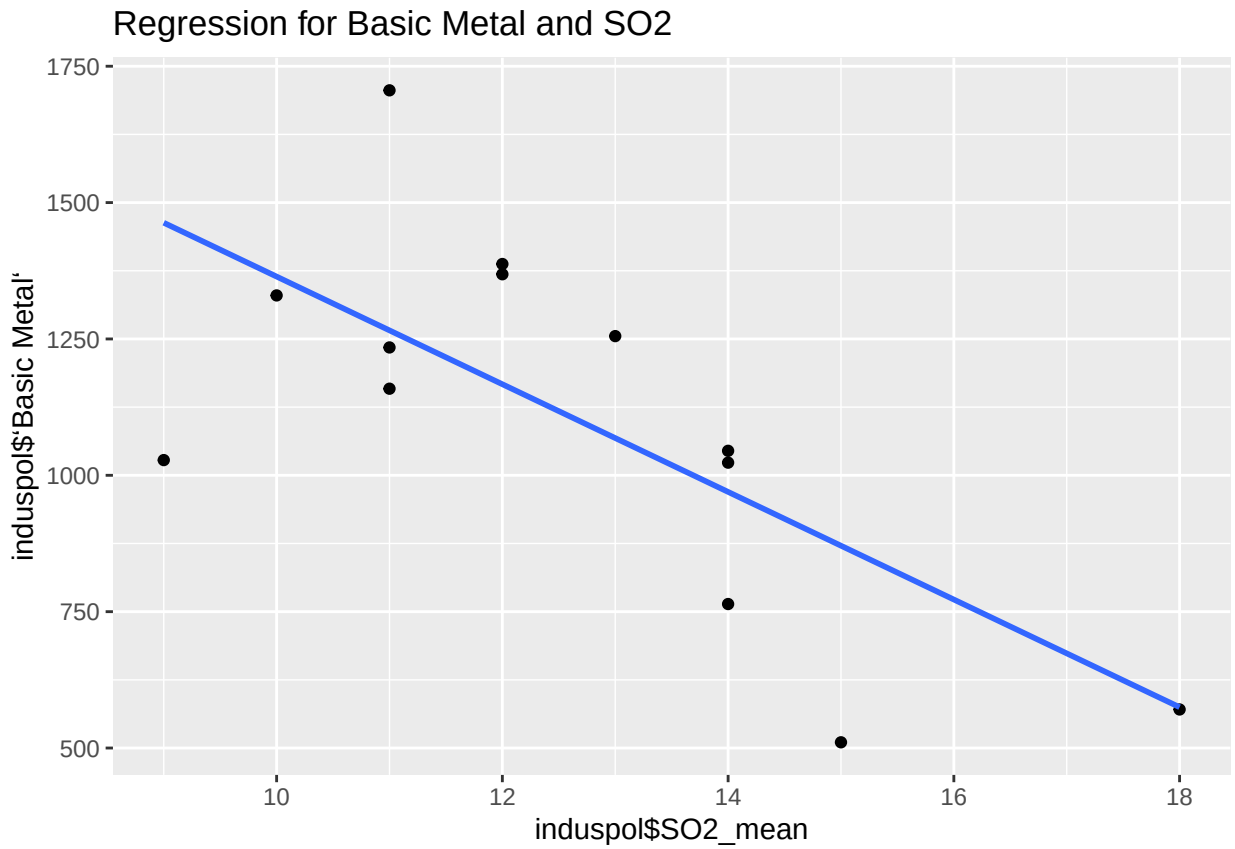
Based on the what was done above, I chose the Industries and the Pollutants that had the strongest correlation above 0.6.

Basic Metal

```
MetalLM <- lm(formula = induspol$`Basic Metal`~ induspol$SO2_mean-1)
summary(MetalLM)
MetalRS <- residuals(MetalLM)
plot(MetalRS)
```



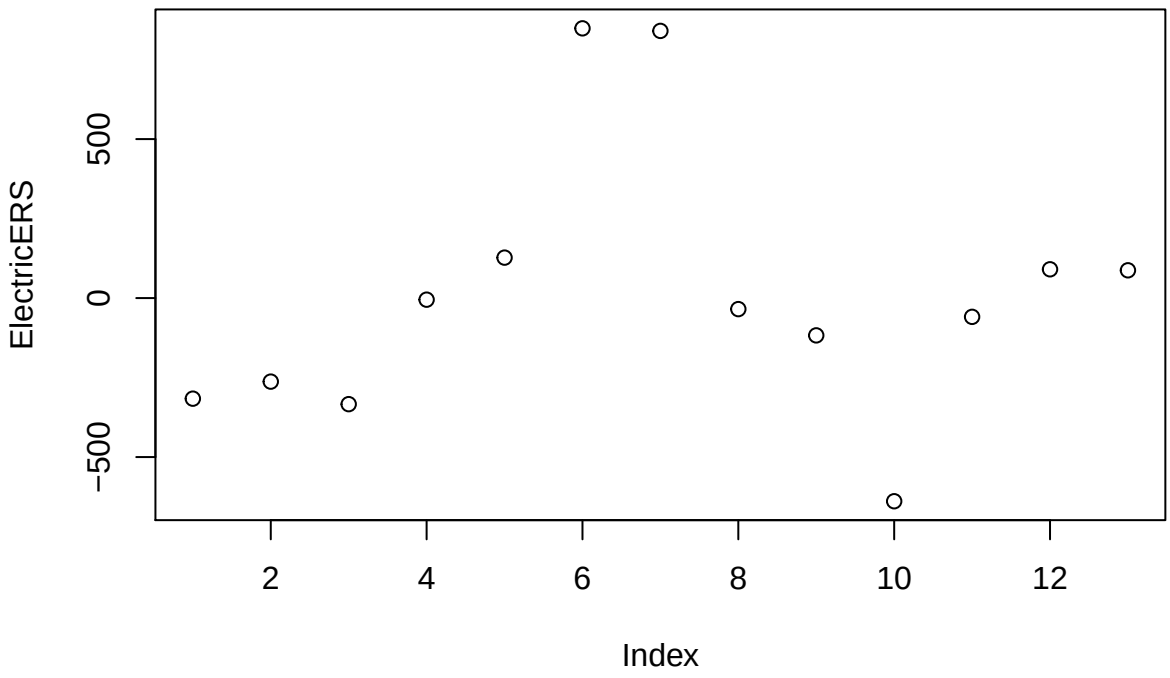

```
ggplot(induspol, aes(x = induspol$SO2_mean, y = induspol$`Basic Metal`)) +
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+
  ggtitle("Regression for Basic Metal and SO2")
```



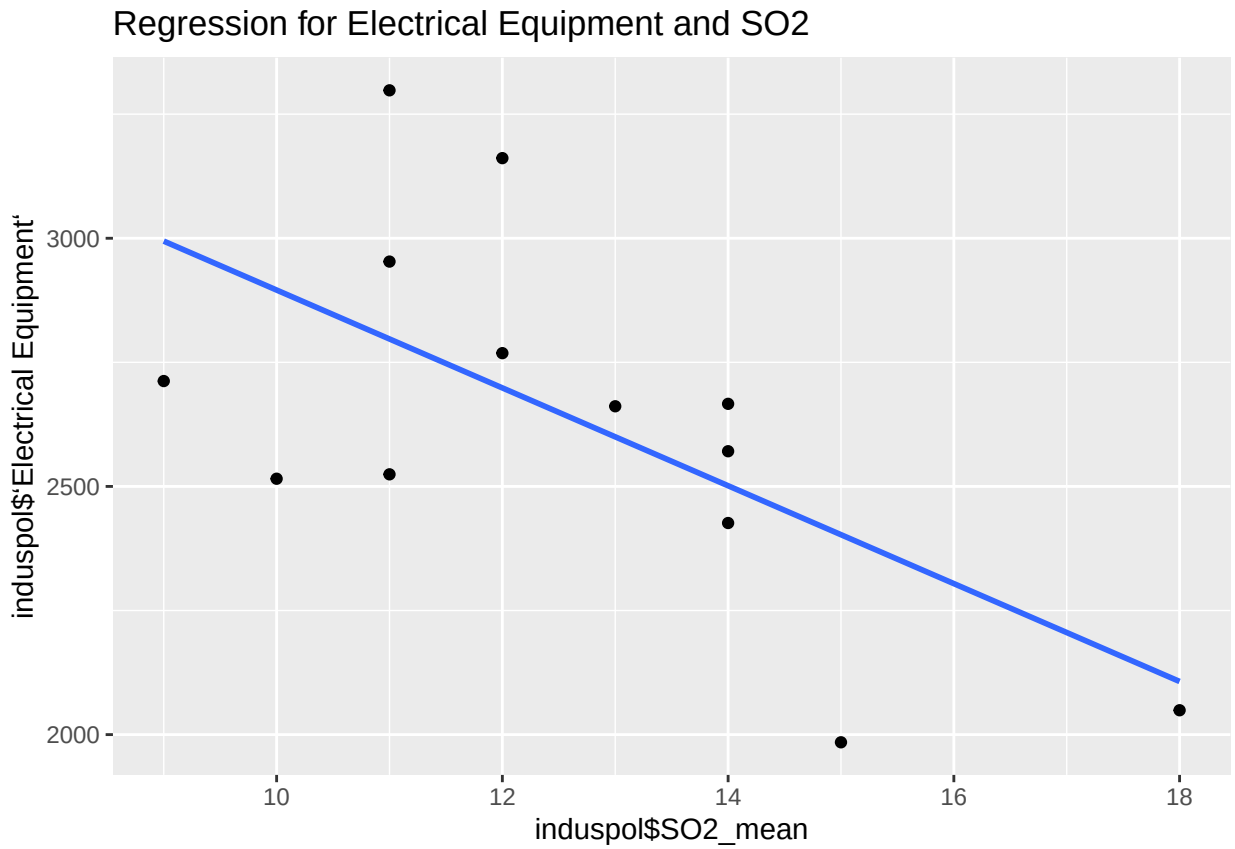

```
##
## Call:
## lm(formula = induspol$`Basic Metal` ~ induspol$SO2_mean - 1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -899.3 -120.2  260.4  388.6  807.5
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## induspol$SO2_mean    81.67      10.86   7.523 7.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 502 on 12 degrees of freedom
## Multiple R-squared:  0.8251, Adjusted R-squared:  0.8105
## F-statistic: 56.59 on 1 and 12 DF,  p-value: 7.015e-06
```

Electric Equipment

```
ElectricELM <-lm(formula = induspol$`Electrical Equipment`~ induspol$NO2_mean+induspol$SO2_mean-1)
summary(ElectricELM)
ElectricERS<-residuals(ElectricELM)
plot(ElectricERS)
```

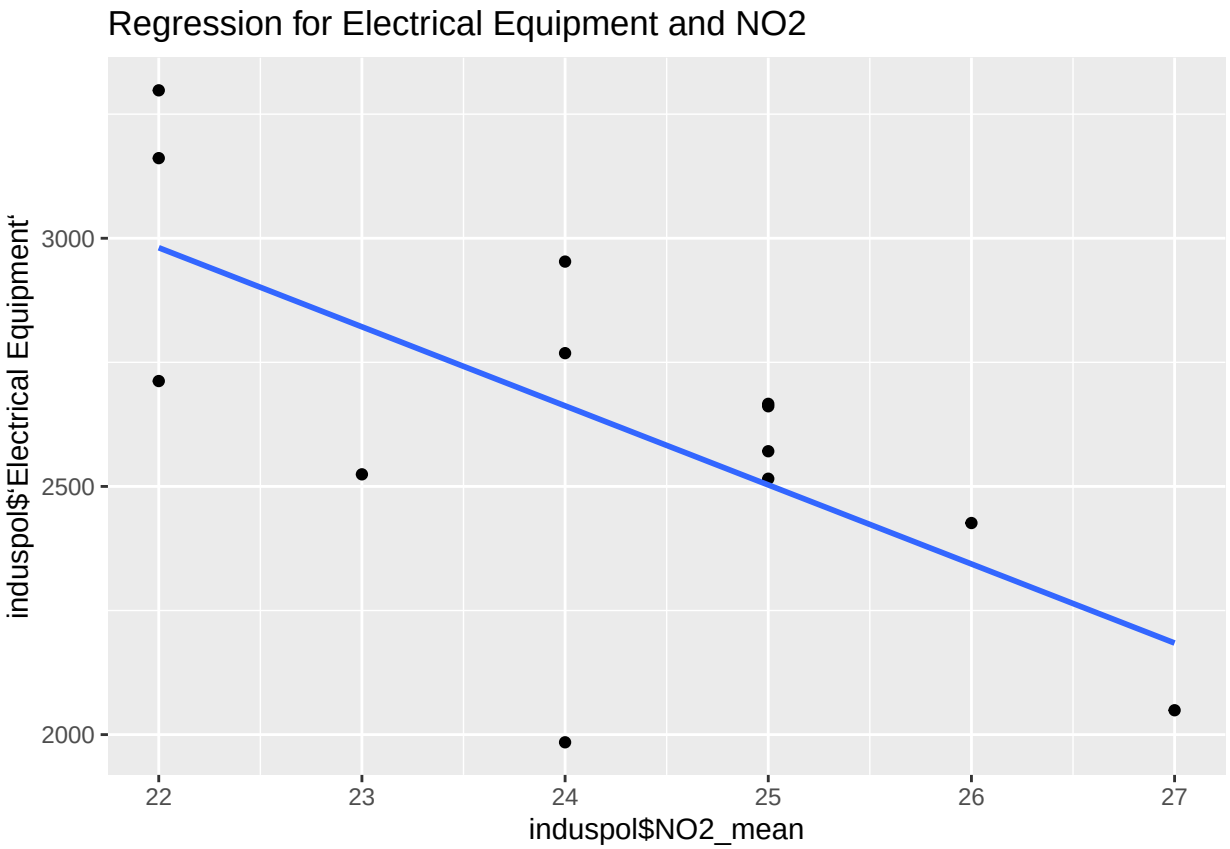



```
ggplot(induspol, aes(x = induspol$SO2_mean, y = induspol$`Electrical Equipment`)) +
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+
  ggtitle("Regression for Electrical Equipment and SO2")
```



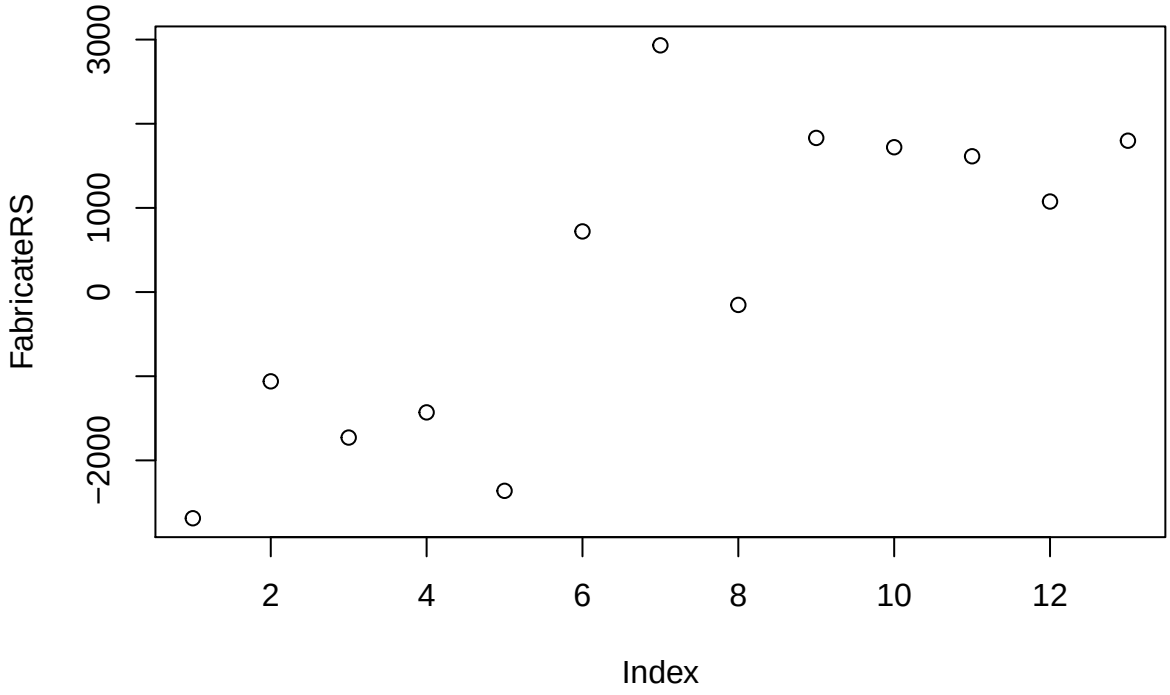

```
##
## Call:
## lm(formula = induspol$`Electrical Equipment` ~ induspol$NO2_mean +
##     induspol$SO2_mean - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -638.97 -262.74  -34.71   90.49  848.31
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## induspol$NO2_mean   184.03      35.89   5.128 0.000329 ***
## induspol$SO2_mean  -144.64      67.73  -2.136 0.056026 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 443.6 on 11 degrees of freedom
## Multiple R-squared:  0.9765, Adjusted R-squared:  0.9722
## F-statistic: 228.6 on 2 and 11 DF,  p-value: 1.098e-09
```

```
ggplot(induspol, aes(x = induspol$NO2_mean, y = induspol$`Electrical Equipment`)) +
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+
  ggtitle("Regression for Electrical Equipment and NO2")
```

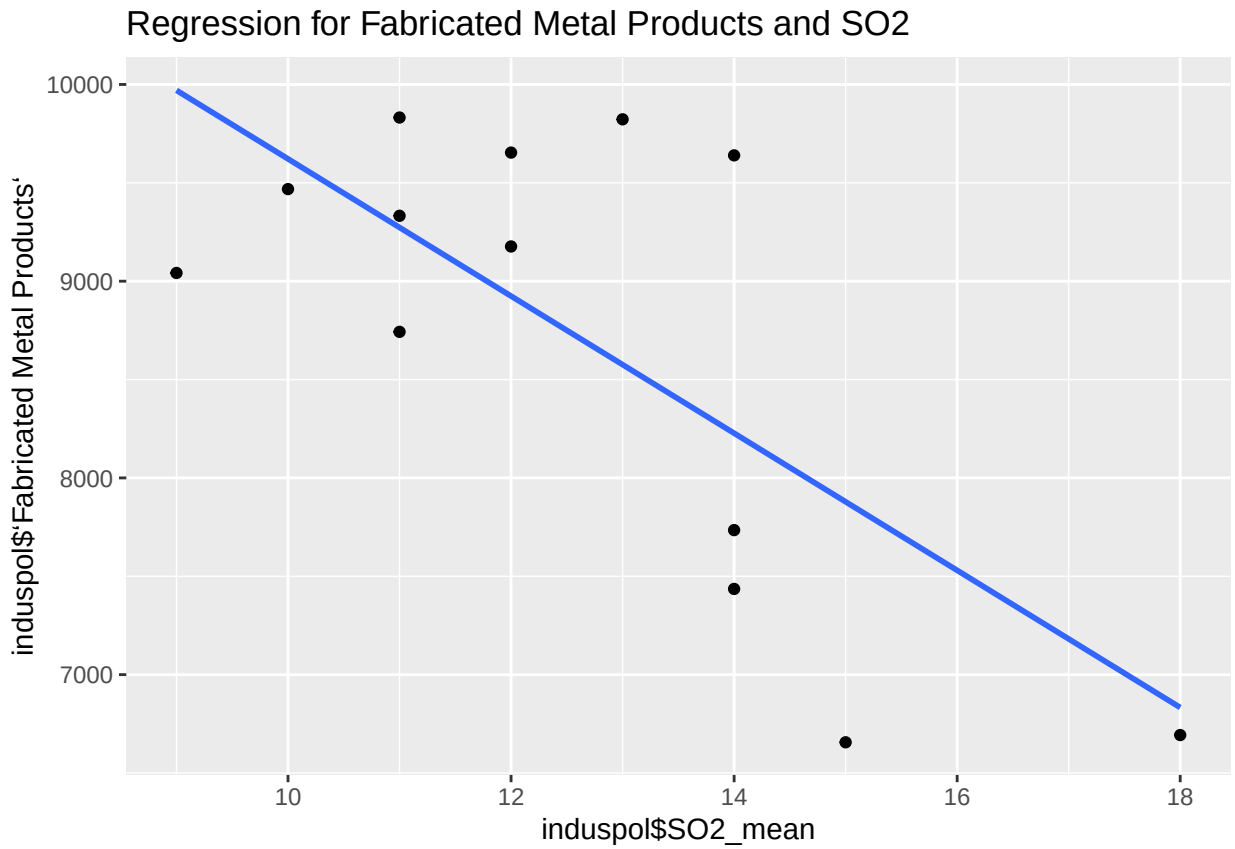


Fabricated Metal Products

```
FabricateLM <- lm(formula = induspol$`Fabricated Metal Products`~ induspol$SO2_mean+induspol$pm2.5_mean
summary(FabricateLM)
FabricateRS <- residuals(FabricateLM)
plot(FabricateRS)
```

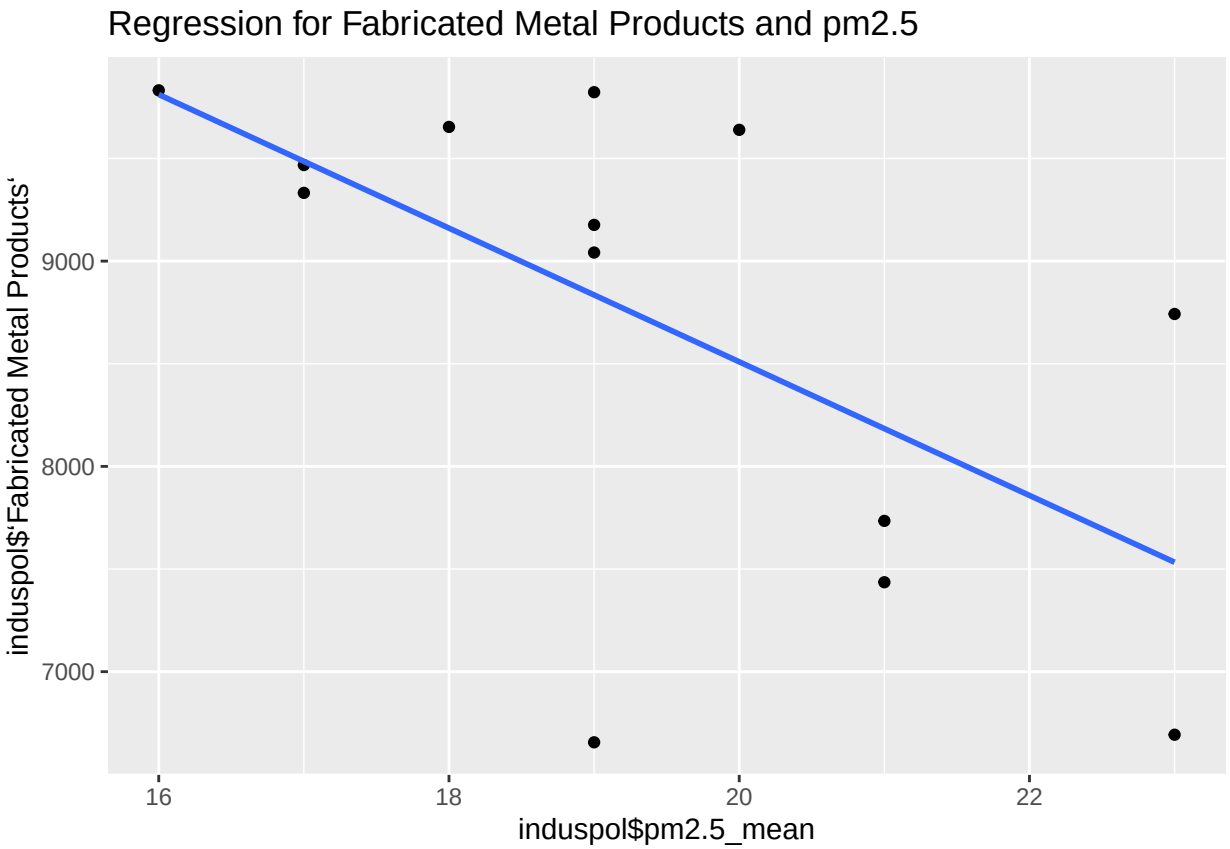



```
ggplot(induspol, aes(x = induspol$SO2_mean, y = induspol$`Fabricated Metal Products`)) +
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+
  ggtitle("Regression for Fabricated Metal Products and SO2")
```



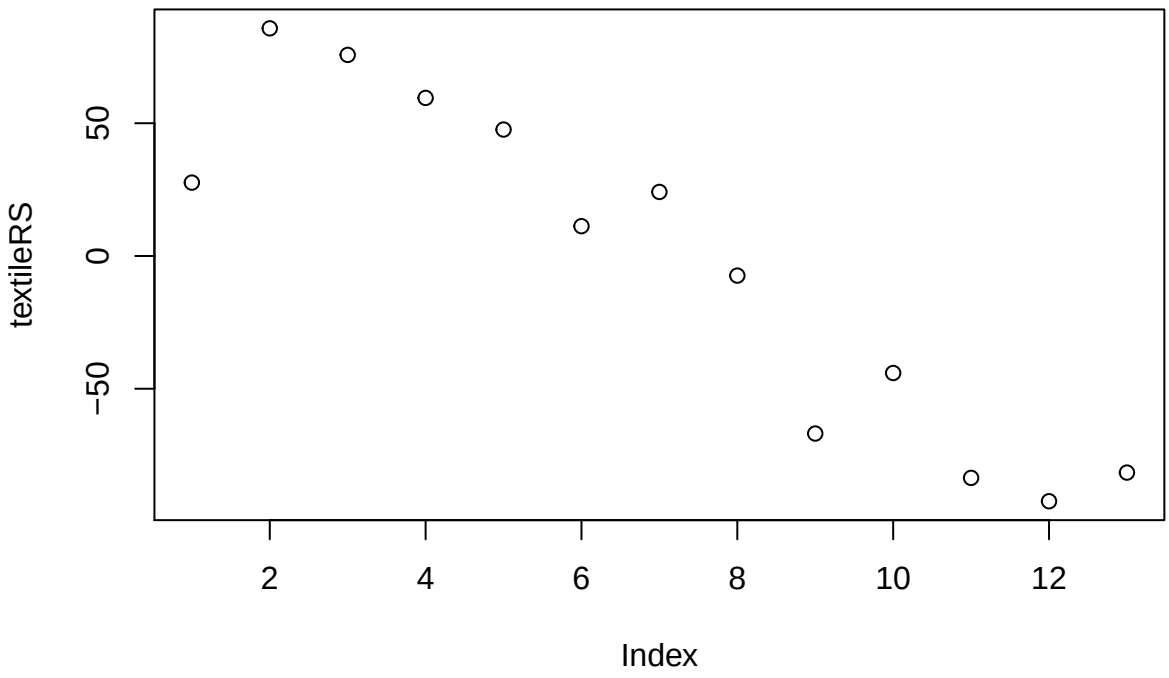

```
##
## Call:
## lm(formula = induspol$`Fabricated Metal Products` ~ induspol$SO2_mean +
##     induspol$pm2.5_mean - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2687.6 -1429.8   720.7  1720.9  2931.6
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## induspol$SO2_mean     -246.3      284.5  -0.866  0.40522
## induspol$pm2.5_mean    600.6      187.2   3.209  0.00832 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1940 on 11 degrees of freedom
## Multiple R-squared:  0.9587, Adjusted R-squared:  0.9512
## F-statistic: 127.7 on 2 and 11 DF,  p-value: 2.442e-08
```

```
ggplot(induspol, aes(x = induspol$pm2.5_mean, y = induspol$`Fabricated Metal Products`)) +
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+
  ggtitle("Regression for Fabricated Metal Products and pm2.5")
```

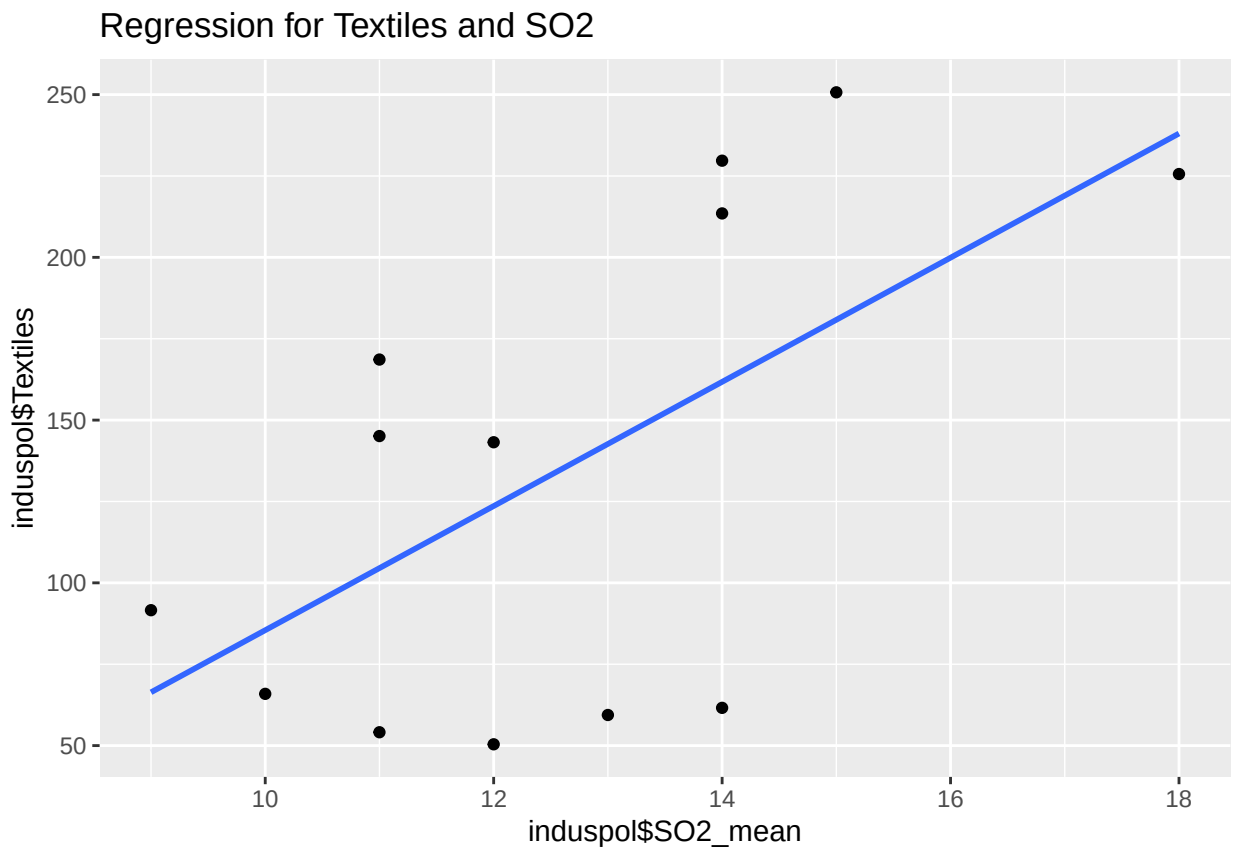


Textiles

```
textileLM <- lm(formula = induspol$Textiles~ induspol$SO2_mean-1)
summary(textileLM)
textileRS <- residuals(textileLM)
plot(textileRS)
```



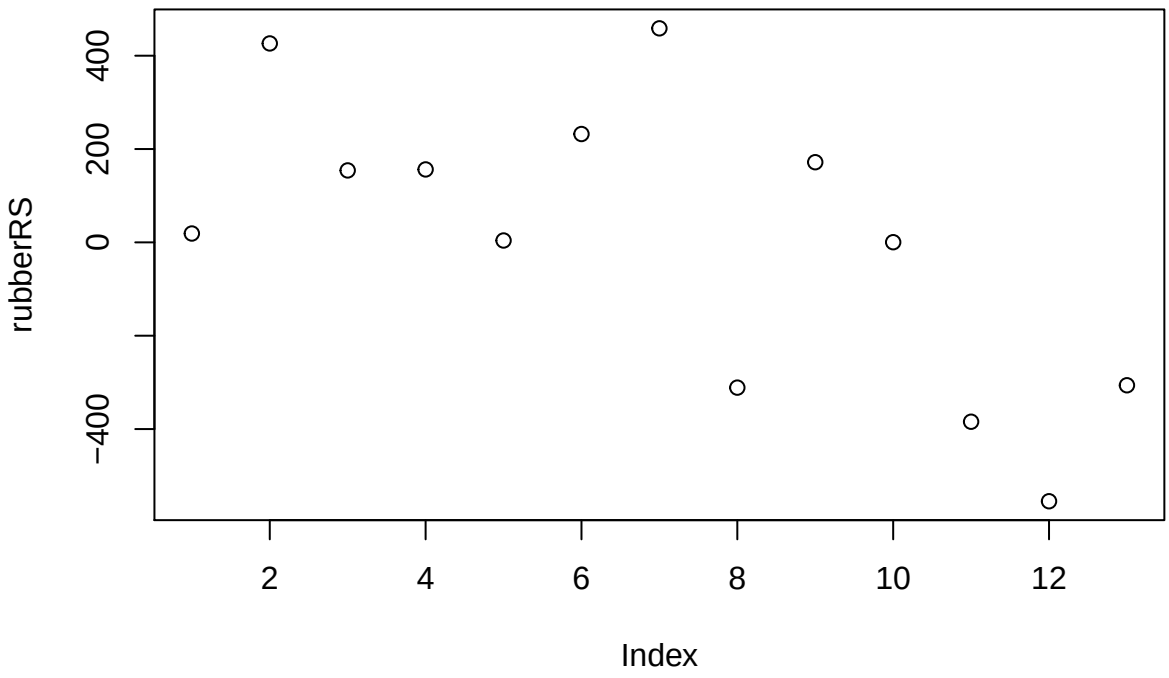

```
ggplot(induspol, aes(x = induspol$SO2_mean, y = induspol$Textiles)) +
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+
  ggtitle("Regression for Textiles and SO2")
```



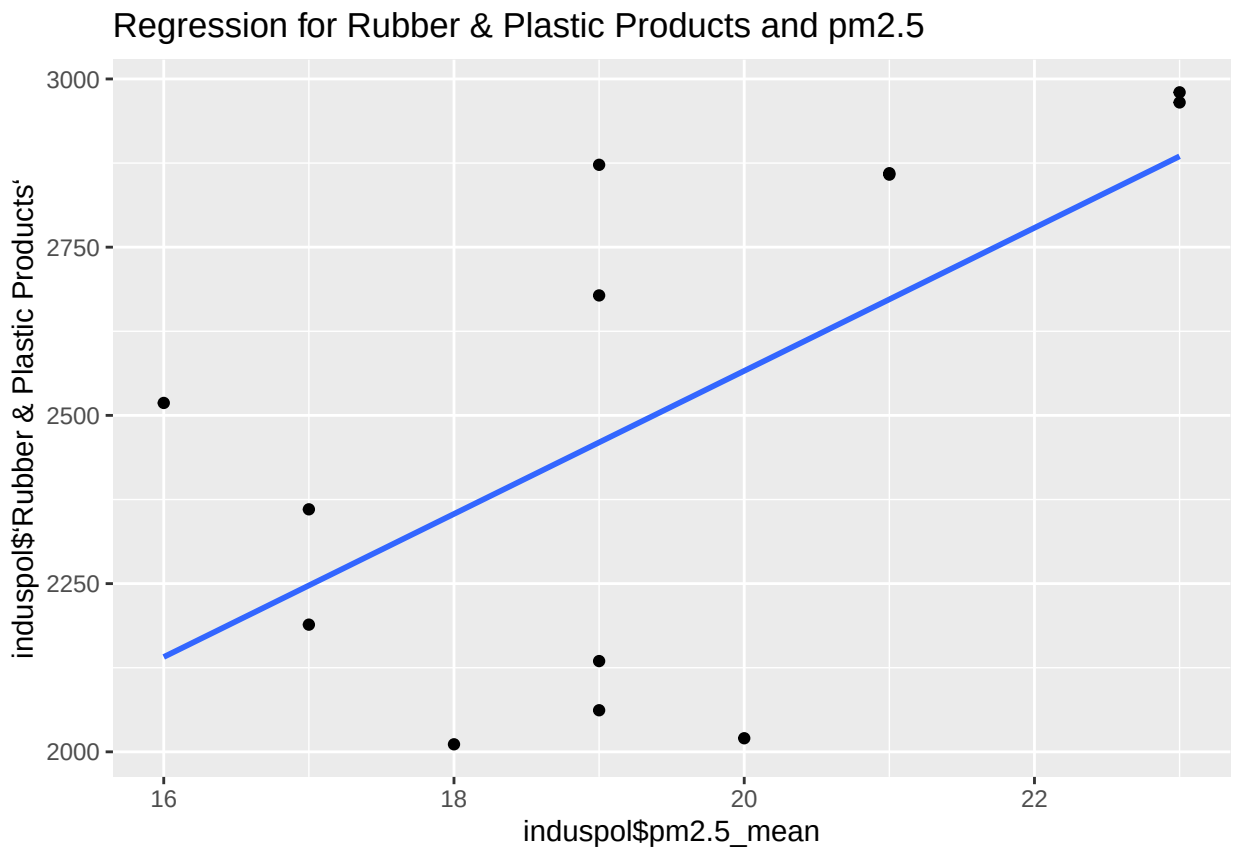

```
##
## Call:
## lm(formula = induspol$Textiles ~ induspol$SO2_mean - 1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -92.37 -66.88  11.23  47.62  85.73
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## induspol$SO2_mean   10.998      1.382   7.957 3.97e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.91 on 12 degrees of freedom
## Multiple R-squared:  0.8407, Adjusted R-squared:  0.8274
## F-statistic: 63.32 on 1 and 12 DF,  p-value: 3.972e-06
```

Rubber & Plastic Products

```
rubberLM <- lm(formula = induspol$`Rubber & Plastic Products`~ induspol$pm2.5_mean-1)
summary(rubberLM)
rubberRS <- residuals(rubberLM)
plot(rubberRS)
```




```
ggplot(induspol, aes(x = induspol$pm2.5_mean, y = induspol$`Rubber & Plastic Products`)) +
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+
  ggtitle("Regression for Rubber & Plastic Products and pm2.5")
```




```
##
## Call:
## lm(formula = induspol$`Rubber & Plastic Products` ~ induspol$pm2.5_mean -
##     1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -554.68 -306.19   18.99  171.75  458.59
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## induspol$pm2.5_mean  128.744      4.442   28.98 1.77e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 312.2 on 12 degrees of freedom
## Multiple R-squared:  0.9859, Adjusted R-squared:  0.9847
## F-statistic: 840.2 on 1 and 12 DF,  p-value: 1.77e-12
```

Regression Overview

So when we look at some of the variables I compared, I see that most of them have a strong a strong dataset. When looking at the P-value of all the F-statistic of all the models made above, all are below 0.05 which shows the overall model is significant. Overall the Adjusted R-squared for most of the models created are above 0.95 with only 2 being above 0.8 which are the Textiles and Basic Metals industry. Nonetheless, we

that the Industry Itself has a strong Impact on some pollutants being significant as we look at the t value in our coefficiants.

The Plan (Endgame)

Scaling and Prepping for Plots

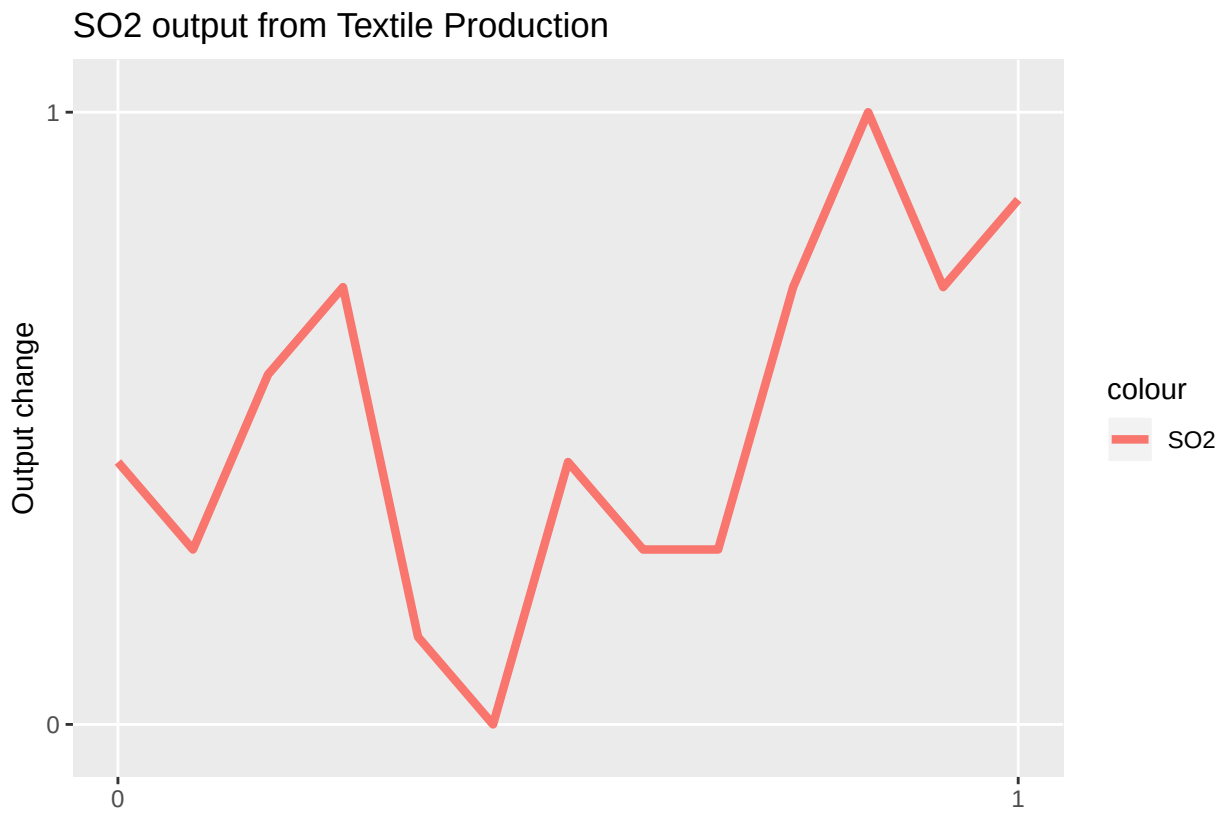
```
scaledinduspoll <- as.data.frame(apply(induspol[,-c(1)], 2, function(x) (x - min(x))/(max(x)-min(x))))

scaledinduspoll$year <- as.factor(induspol$year)
scaledinduspoll$SO2_mean <- as.factor(scaledinduspoll$SO2_mean)
scaledinduspoll$NO2_mean <- as.factor(scaledinduspoll$NO2_mean)
scaledinduspoll$pm2.5_mean <- as.factor(scaledinduspoll$pm2.5_mean)
scaledinduspoll$`Basic Metal` <- as.factor(scaledinduspoll$`Basic Metal`)
scaledinduspoll$`Electrical Equipment` <- as.factor(scaledinduspoll$`Electrical Equipment`)
scaledinduspoll$`Fabricated Metal Products` <- as.factor(scaledinduspoll$`Fabricated Metal Products`)
scaledinduspoll$Textiles <- as.factor(scaledinduspoll$Textiles)
scaledinduspoll$`Rubber & Plastic Products`<- as.factor(scaledinduspoll$`Rubber & Plastic Products`)
```

This is just more or less scaling my dataset again. When I scaled previously, it was only pollutants. This dataset has Industry output also, so I need to scale it again to plot.

Last Plots

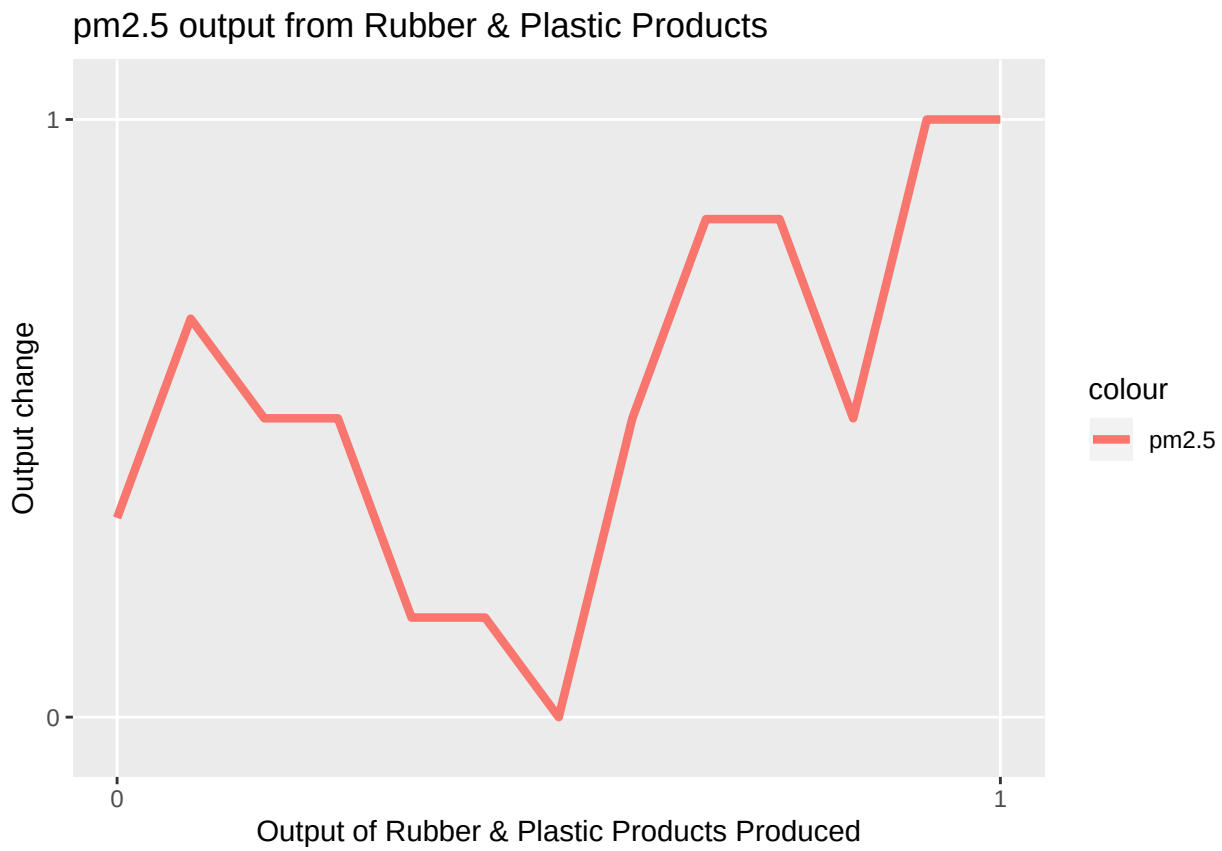
Positive Correlations

```
ggplot(scaledinduspoll, aes(x = scaledinduspoll$Textiles, group = 1)) +
  geom_line(aes(y=scaledinduspoll$SO2_mean, colour = "SO2"), size = 1.5) +
  scale_y_discrete(breaks = seq(from=0, to=1, by=0.1)) +
  ggtitle("SO2 output from Textile Production") +
  scale_x_discrete(breaks = seq(from=0, to=1, by=0.1)) +
  ylab("Output change")+
  xlab("")
```



For the amount of Textiles produced, the amount of SO2 outputted is about the same.

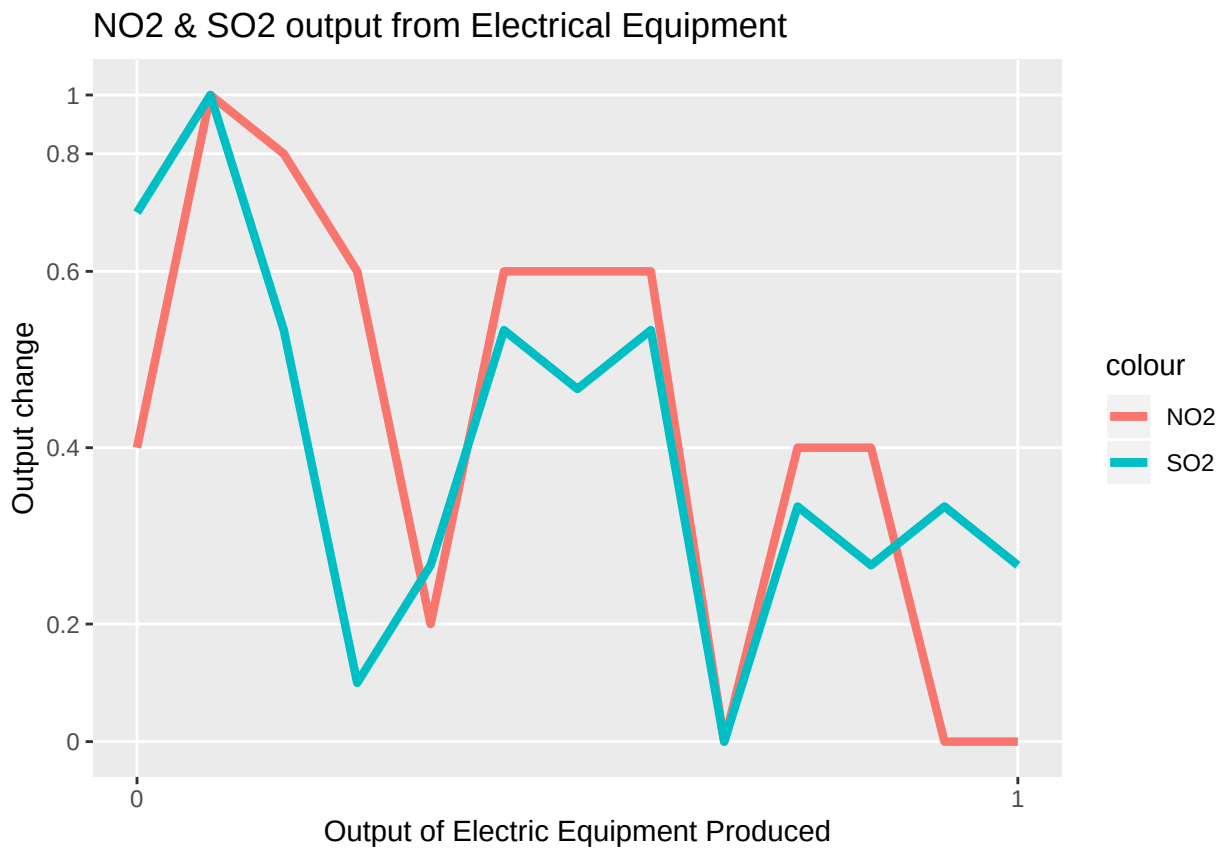
```
ggplot(scaledinduspoll, aes(x = scaledinduspoll$`Rubber & Plastic Products`, group = 1)) +
  geom_line(aes(y = scaledinduspoll$pm2.5_mean,
                colour = "pm2.5"), size = 1.5) +
  scale_y_discrete(breaks = seq(from=0, to=1, by=0.1)) +
  ggtitle("pm2.5 output from Rubber & Plastic Products") +
  scale_x_discrete(breaks = seq(from=0, to=1, by=0.1)) +
  ylab("Output change")+
  xlab("Output of Rubber & Plastic Products Produced")
```



For the amount of Rubber & plastic Produced, the amount of pm2.5 in the air is about the same.

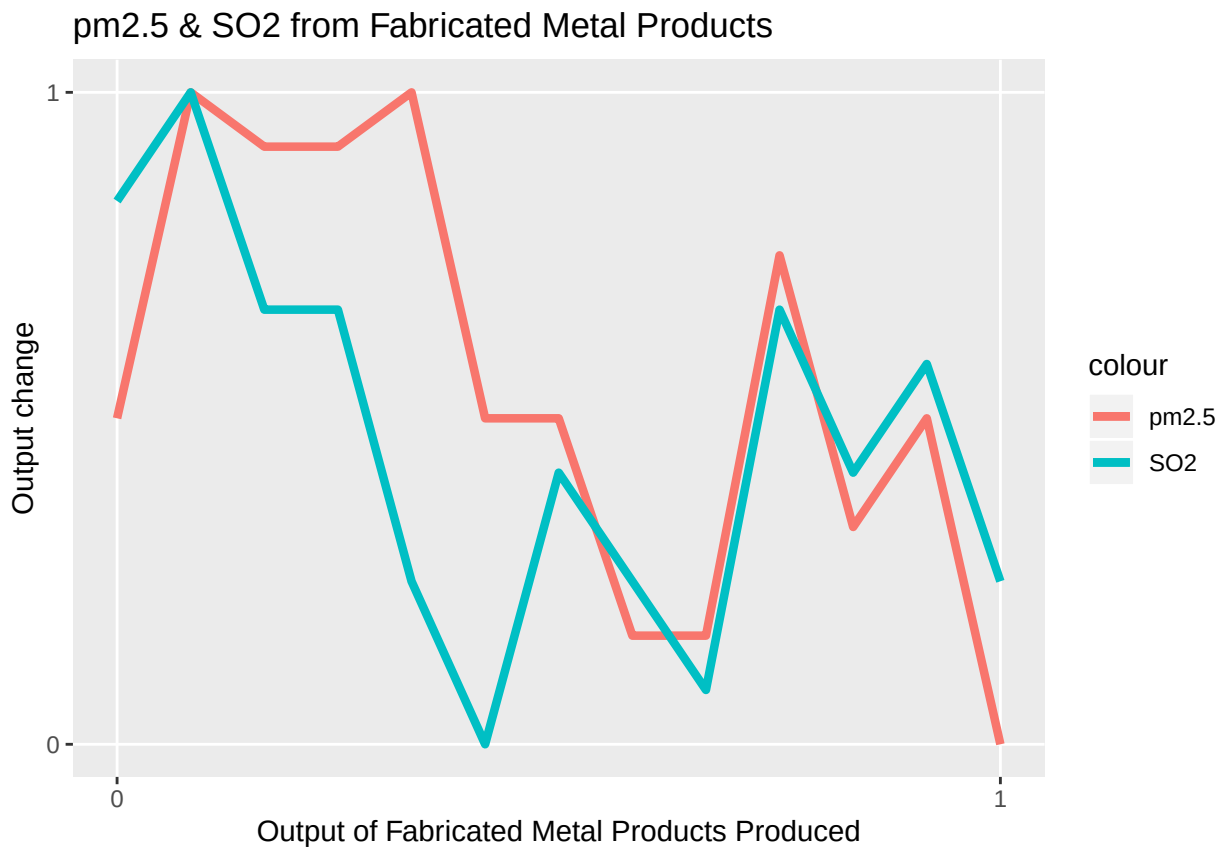
Negative Correlations

```
ggplot(scaledinduspoll, aes(x = scaledinduspoll$`Electrical Equipment`, group = 1)) +
  geom_line(aes(y = scaledinduspoll$NO2_mean, colour = "NO2"), size = 1.5) +
  geom_line(aes(y = scaledinduspoll$SO2_mean, colour = "SO2"), size = 1.5) +
  scale_y_discrete(breaks = seq(from=0, to=1, by=0.1)) +
  ggtitle("NO2 & SO2 output from Electrical Equipment") +
  scale_x_discrete(breaks = seq(from=0, to=1, by=0.1)) +
  ylab("Output change")+
  xlab("Output of Electric Equipment Produced")
```



For the amount of Electric Equipment Produced, the amount of NO2 and SO2 outputted is decreasing.

```
ggplot(scaledinduspoll, aes(x = scaledinduspoll$`Fabricated Metal Products`, group = 1)) +
  geom_line(aes(y = scaledinduspoll$pm2.5_mean, colour = "pm2.5"), size = 1.5) +
  geom_line(aes(y = scaledinduspoll$SO2_mean, colour = "SO2"), size = 1.5) +
  scale_y_discrete(breaks = seq(from=0, to=1, by=0.1)) +
  ggtitle("pm2.5 & SO2 from Fabricated Metal Products") +
  scale_x_discrete(breaks = seq(from=0, to=1, by=0.1)) +
  ylab("Output change")+
  xlab("Output of Fabricated Metal Products Produced")
```



For the amount of Fabricated Metal products produced, the amount of SO2 outputted and pm2.5 in the air is decreasing.

Conclusion

Results

Based on the results of my data I can conclude that, for certain industries, the amount of pollutants outputted varies on if i reject or accept the null hypothesis. For example, when looking at Textiles and Rubber & Plastic Products, I would accept the Null Hypothesis because the output of some pollutants haven't changed. Then when we look Electric Equipment and Fabricated Metal Products, we see a decrease in some of the pollutants outputted. This could be because the efficiency throughout the years has impacted the output of pollutants as more is produced. Sometimes the industry itself doesn't have a big enough impact at all on other Pollutants.

Thoughts and complications

As you might have saw in the datasets I read in, I havent used a few. This was due to the time constraints I had at the time and couldnt do more an in depth analysis of industries and other things that can contribute to the pollution in Singapore. When it came to the dataset I was working on, my datasets was reduced in the number of observations. This is because the data I orginally used only had recorded pollutants starting from specific years such as 2002. For now the assumption I'm going with is that people in the past might not have recorded it, or didnt have the technology available at that time. However the data was really reliable as

shown from the Regression section. I will probably push this to my personal github as a cool project I can coninue to work on in Python, my programming language of choice, for further analysis in the future.

Sources that are helping me understand my dataset

- https://www.epa.gov/ground-level-ozone-pollution/ground-level-ozone-basics
- http://www.npi.gov.au/resource/particulate-matter-pm10-and-pm25
- https://www.environment.gov.au/protection/publications/factsheet-nitrogen-dioxide-no2
- https://www.dhs.wisconsin.gov/chemical/sulfurdioxide.htm
- https://www.cdc.gov/co/faqs.htm
- https://en.wikipedia.org/wiki/Design,_Build_and_Sell_Scheme
- https://www.hdb.gov.sg/cs/infoweb/homepage
- https://www.youtube.com/watch?v=49fADBfcDD4 (ggplot tutorial)
- https://www.youtube.com/watch?v=lTTJPRwnONE&list=PLLxj8fULvXwGOf8uHlL4Tr62oXSB5k_ in (TidyVerse Tutorial. watched a few from here)
- https://www.youtube.com/watch?v=1ahg7h5DRP4
- https://stackoverflow.com/questions/41020820/plot-linear-regression-on-multiple-columns-within-specific-range
- https://stackoverflow.com/questions/47058761/extract-all-rows-with-any-value-greater-than-x
- https://stackoverflow.com/questions/14096814/merging-a-lot-of-data-frames
- https://stackoverflow.com/questions/11433432/how-to-import-multiple-csv-files-at-once
- https://medium.com/@swethalakshmanan14/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f
- https://stats.stackexchange.com/questions/210525/what-is-zero-mean-and-unit-variance-in-terms-of-image-data
- https://datascience.stackexchange.com/questions/32109/zero-mean-and-unit-variance
- https://www.youtube.com/watch?v=u7TxjUI4PRI (Regression)
- https://www.youtube.com/watch?v=sKW2umonEvY (Regression)

  I couldnt really find papers to help on this dataset only because of the many variables that are involved. To represent a dataset first I had to know what the data I was given meant. When searching up anything, I would search mostly of how I wanted to manipulate my data and get what I want from them. Websites which give these small burst of information on one variable at a time helps me get a quick overview of what to expect as I do more. Stackoverflow was a huge help and watching some Youtube Tutorials helped me do what I wanted by my own methods. Alot of what the resources I've listed here show snippets of code that I resued for my own pupose, along with explanations I needed to know what was going on.