Air Quality Analysis in India

Authors

- Mahendra Gajula Pothamsetty, magj7729@colorado.edu
- Manasa Kolavennu, veko4270@colorado.edu
- Vishadh Vilas Sawant, visa4353@colorado.edu
- Santosh Adabala, siad4700@colorado.edu
- Pugazhventhan Kulandaivel, puku4269@colorado.edu

Introduction

Environmental contamination has become a growing concern since industrialization. Air pollution is the biggest environmental concern in the world, according to a WHO analysis that links air pollution to 7 million early deaths each year. Furthermore, it has been discovered that India's air pollution is deadlier than even China's, as revealed in the NY Times article, India's Air Pollution Rivals China's as World's Deadliest.

Goal of the project:

Can we recognize regional trends? Can we link changes in environmental policy in India to changes in air quality?

Initial Setup:

Let's start with importing few handy libraries like tidyverse, ggplot2, tidyr, dplyr, stringr for data manipulation, data correction and visualizations

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## v purrr   0.3.5
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidyr)
library(dplyr)
library(stringr)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

About the Dataset:

Under the National Data Sharing and Accessibility Policy, the Ministry of Environment and Forests and the Central Pollution Control Board of India issued an unified and Historical Daily Ambient Air Quality Data (NDSAP) dataset. This dataset contains the attributes causing pollution like SO2(Sulphur Dioxide), NO2 (Nitrogen Dioxide), rspm (Respirable Suspended Particulate Matter) and in addition we have information regarding the geo-logical location where the air sample is tested like state, location, location_monitoring_station along with the date on which data is collected.

This dataset allows for a more detailed examination of India's air pollution levels.

Dataset is uploaded on GitHub, hence it must be read from github and a little glimpse of data is displayed.

```
url = 'https://raw.githubusercontent.com/mahendra-g-p/DataScience_as_a_Field_Project/main/Data/data.csv


pollution_data <- readr::read_csv(url)
```

```
## Rows: 435742 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): stn_code, sampling_date, state, location, agency, type, location_m...
## dbl  (5): so2, no2, rspm, spm, pm2_5
## date (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(pollution_data)
```

```
## # A tibble: 6 x 13
##   stn_code sampling~1 state locat~2 agency type    so2   no2  rspm   spm locat~3
##   <chr>    <chr>      <chr> <chr>   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <chr>
## 1 150      February ~ Andh~ Hydera~ <NA>   Resi~   4.8  17.4    NA    NA <NA>
## 2 151      February ~ Andh~ Hydera~ <NA>   Indu~   3.1   7      NA    NA <NA>
## 3 152      February ~ Andh~ Hydera~ <NA>   Resi~   6.2  28.5    NA    NA <NA>
## 4 150      March - M~ Andh~ Hydera~ <NA>   Resi~   6.3  14.7    NA    NA <NA>
## 5 151      March - M~ Andh~ Hydera~ <NA>   Indu~   4.7   7.5    NA    NA <NA>
## 6 152      March - M~ Andh~ Hydera~ <NA>   Resi~   6.4  25.7    NA    NA <NA>
## # ... with 2 more variables: pm2_5 <dbl>, date <date>, and abbreviated variable
## #   names 1: sampling_date, 2: location, 3: location_monitoring_station
```

Data Cleaning:

Firstly, total null values in the dataset need to be evaluated for each column.

```
colSums(is.na(pollution_data))
```

```
##                    stn_code               sampling_date
##                      144077                           3
##                       state                    location
##                           0                           3
```

```
##                       agency                         type
##                       149481                         5393
##                          so2                          no2
##                        34646                        16233
##                         rspm                          spm
##                        40222                       237387
## location_monitoring_station                         pm2_5
##                        27491                       426428
##                         date
##                            7
```

It is evident that data in columns like stn_code, agency, spm, pm2_5 are having null values more than the permissable levels (Considered threshold level as 10%). As the percentage of null values is very huge, it is better to remove those columns. Hence the above mentioned columns are removed.

```
pollution_data = subset(pollution_data, select = -c(stn_code, agency, spm, pm2_5))

pollution_data
```

```
## # A tibble: 435,742 x 9
##    sampling_date      state   locat~1 type    so2   no2  rspm locat~2 date
##    <chr>              <chr>   <chr>   <chr> <dbl> <dbl> <dbl> <chr>   <date>
##  1 February - M021990 Andhra~ Hydera~ Resi~   4.8  17.4    NA <NA>    1990-02-01
##  2 February - M021990 Andhra~ Hydera~ Indu~   3.1   7      NA <NA>    1990-02-01
##  3 February - M021990 Andhra~ Hydera~ Resi~   6.2  28.5    NA <NA>    1990-02-01
##  4 March - M031990    Andhra~ Hydera~ Resi~   6.3  14.7    NA <NA>    1990-03-01
##  5 March - M031990    Andhra~ Hydera~ Indu~   4.7   7.5    NA <NA>    1990-03-01
##  6 March - M031990    Andhra~ Hydera~ Resi~   6.4  25.7    NA <NA>    1990-03-01
##  7 April - M041990    Andhra~ Hydera~ Resi~   5.4  17.1    NA <NA>    1990-04-01
##  8 April - M041990    Andhra~ Hydera~ Indu~   4.7   8.7    NA <NA>    1990-04-01
##  9 April - M041990    Andhra~ Hydera~ Resi~   4.2  23      NA <NA>    1990-04-01
## 10 May - M051990      Andhra~ Hydera~ Indu~   4     8.9    NA <NA>    1990-05-01
## # ... with 435,732 more rows, and abbreviated variable names 1: location,
## #   2: location_monitoring_station
```

It is also observed that there are null values in columns like type & location_monitoring_station which are categorical attributes. As it is not correct to replace those null values with some random value, rows containing those null values are excluded.

```
pollution_data <- pollution_data[!(is.na(pollution_data$type) | pollution_data$type==""  | is.na(polluti
```

Now, lets see how many null values exist in each column

```
colSums(is.na(pollution_data))
```

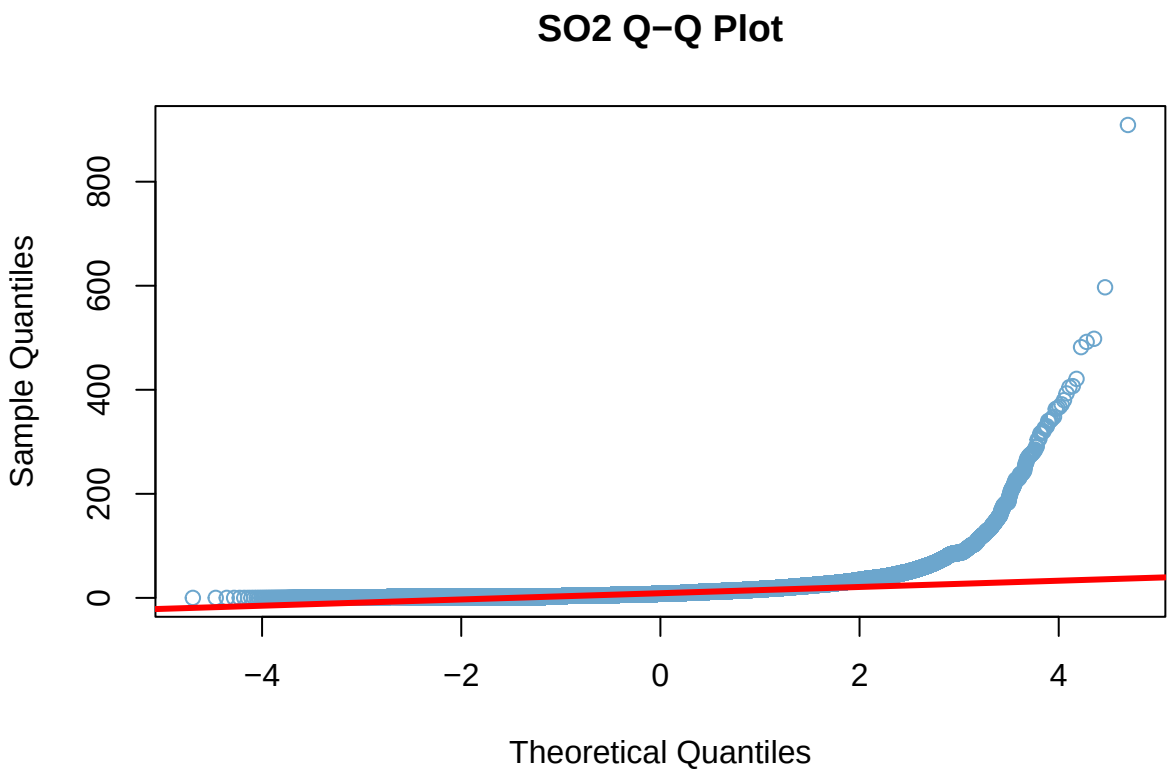
```
##               sampling_date                      state
##                           0                          0
##                    location                       type
##                           0                          0
##                         so2                        no2
##                       32657                      14736
##                        rspm location_monitoring_station
```

```
##                         14432                             0
##                          date
##                             4
```

Finally, we are left with null values in numerical attributes i.e., so2, no2 & rspm

Let's see the distribution of SO2 data

```
qqnorm(pollution_data$so2, col = "skyblue3",
       main = "SO2 Q-Q Plot")
qqline(pollution_data$so2, col = "red", lwd = 3)
```

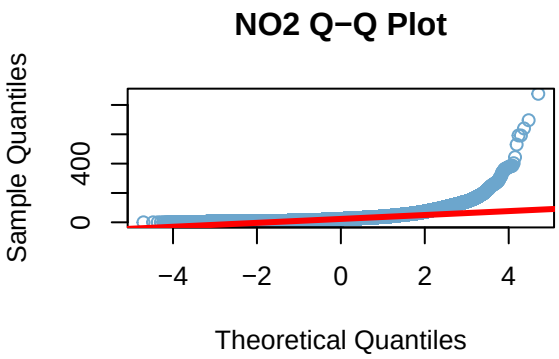


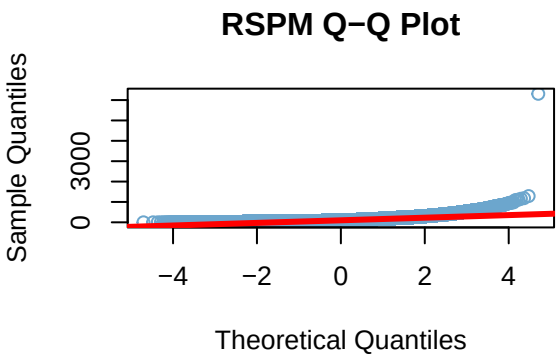
It seems the data in SO2 is right-skewed. Now let's find out the distributions of NO2 & RSPM data as well.

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

qqnorm(pollution_data$no2, col = "skyblue3",
       main = "NO2 Q-Q Plot")
qqline(pollution_data$no2, col = "red", lwd = 3)


qqnorm(pollution_data$rspm, col = "skyblue3",
       main = "RSPM Q-Q Plot")
qqline(pollution_data$rspm, col = "red", lwd = 3)
```





It seems to be the data in NO2 & RSPM columns are also right-skewed.

First let's take so2 attribute for replacing the null values. As the data is skewed, it is better to replace the null values with median instead of mean.

Now, the question is which mean we need to consider. Is it the mean of entire so2 column data?

It would be not right as there is data from multiple locations. So better to replace the null values of a location by the median of it's respective location.

For that, let's subset the entire data with just location and so2 columns then group the data by location and finally add a column with median of so2 values for that location.

```
subset_so2 <- pollution_data %>% select(location,so2) %>% group_by(location) %>% summarize(value = medi
sum(is.na(subset_so2$value))
```

```
## [1] 2
```

```
head(subset_so2)
```

```
## # A tibble: 6 x 2
##   location  value
##   <chr>     <dbl>
## 1 Agra        4.8
## 2 Ahmedabad  13
## 3 Aizawl      2
## 4 Akola       8
```

```
## 5 Alappuzha   2
## 6 Allahabad   4
```

Now the job is to find out null values under each location and replace it with the respective location's median.

```
sum(is.na(pollution_data$so2))
```

```
## [1] 32657
```

```
p <- 0
for (each in c(is.na(pollution_data$so2))) {
  p <- p+1
    if (each == TRUE) {
      h <- subset_so2[match(pollution_data$location[p],subset_so2$location),'value']
      pollution_data$so2[p] <- h[[1]]
      }
}
sum(is.na(pollution_data$so2))
```

```
## [1] 61
```

Almost 99% null values in so2 are removed. Now let's repeat the same replacing process for no2 as the data is not normally distributed. For that, again a subset of data containing unique locations and their respective median values of no2 are calculated.

```
subset_no2 <- pollution_data %>% select(location,no2) %>% group_by(location) %>% summarize(value = media
sum(is.na(subset_no2$value))
```

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

```
head(subset_no2)
```

```
## # A tibble: 6 x 2
##   location  value
##   <chr>     <dbl>
## 1 Agra       17
## 2 Ahmedabad  21
## 3 Aizawl      6
## 4 Akola       9
## 5 Alappuzha   4.5
## 6 Allahabad  26.1
```

Now, replace the null values in no2 under each location with it's respective location's median value.

```
sum(is.na(pollution_data$no2))
```

```
## [1] 14736
```

```
p <- 0
for (each in c(is.na(pollution_data$no2))) {
  p <- p+1
  if (each == TRUE) {
    h <- subset_no2[match(pollution_data$location[p],subset_no2$location),'value']
    pollution_data$no2[p] <- h[[1]]
  }
}
sum(is.na(pollution_data$no2))
```

```
## [1] 45
```

Almost all the null values are removed. Now only null values in rspm column are left. First, let's find out the median of rspm value for each and every unqiue locations.

```
subset_rspm <- pollution_data %>% select(location,rspm) %>% group_by(location) %>% summarize(value = me
sum(is.na(subset_rspm$value))
```

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

```
head(subset_rspm)
```

```
## # A tibble: 6 x 2
##   location  value
##   <chr>     <dbl>
## 1 Agra        166
## 2 Ahmedabad    86
## 3 Aizawl       39
## 4 Akola       132
## 5 Alappuzha    43
## 6 Allahabad   232
```

Now let's replace the null values under each location with it's respective median value.

```
sum(is.na(pollution_data$rspm))
```

```
## [1] 14432
```

```
p <- 0
for (each in c(is.na(pollution_data$rspm))) {
  p <- p+1
  if (each == TRUE) {
    h <- subset_rspm[match(pollution_data$location[p],subset_rspm$location),'value']
    pollution_data$rspm[p] <- h[[1]]
  }
}
sum(is.na(pollution_data$rspm))
```

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

After replacing all those null values, let's check if any null values exist

```
colSums(is.na(pollution_data))
```

```
##                sampling_date                        state
##                            0                            0
##                     location                         type
##                            0                            0
##                          so2                          no2
##                           61                           45
##                         rspm location_monitoring_station
##                            0                            0
##                         date
##                            4
```

Surprisingly, a very small amount of null values are left over in so2 & no2 columns. Better to omit those rows containing null values.

```
pollution_data <- na.omit(pollution_data)
```

Now, let's have a final check on the null values count

```
colSums(is.na(pollution_data))
```

```
##                sampling_date                        state
##                            0                            0
##                     location                         type
##                            0                            0
##                          so2                          no2
##                            0                            0
##                         rspm location_monitoring_station
##                            0                            0
##                         date
##                            0
```

Finally, the data cleaning is done.

Data Manipulations:

Before going for data visualizations, let's breakdown the date into year, month and date.

```
pollution_data = separate(pollution_data, date, into = c('Year','Month','Day'), sep = '-')
pollution_data <- pollution_data %>% mutate_at(c('Year','Month','Day'), as.numeric)
```

Now find out the unique values in type column

```
unique(pollution_data$type)
```

```
## [1] "Industrial Area"                    "Residential, Rural and other Areas"
## [3] "Sensitive Area"                     "Industrial Areas"
## [5] "Residential and others"             "Sensitive Areas"
## [7] "RIRUO"
```

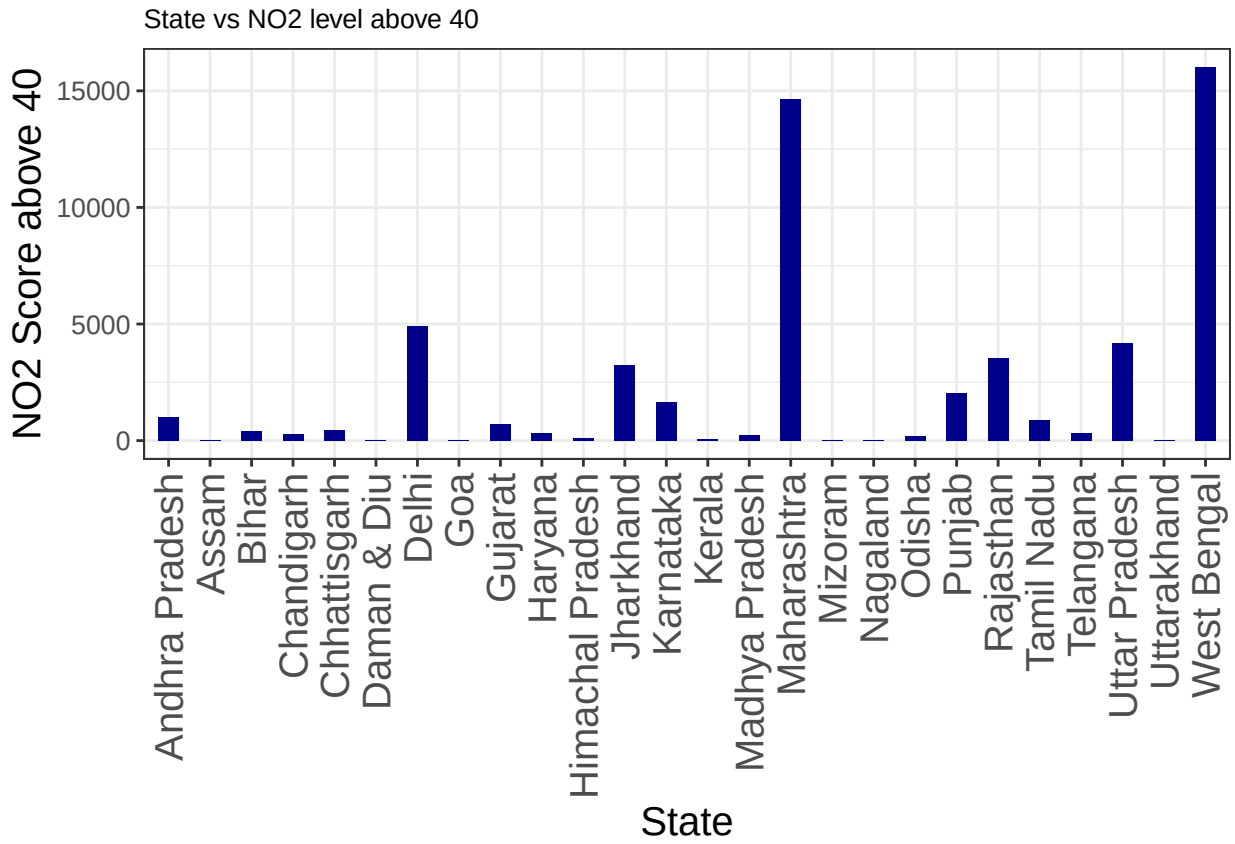
Seems to be some types are pretty much similar by discription but are having various descriptions. So let's reduce the types. For example, replace "Industrial","Industrial Areas" with "Industrial Area"

```
pollution_data$type[pollution_data$type=="Sensitive Areas"] <-"Sensitive Area"
pollution_data$type[pollution_data$type %in% c("Industrial","Industrial Areas")] <-"Industrial Area"
pollution_data$type[pollution_data$type %in% c("Residential and others")] <-"Residential Area"
pollution_data$type[pollution_data$type %in% c("Residential, Rural and other Areas")] <-"RIRUO"
```

Let's explore some data visualizations

Let's start with the NO2 pollutant. According to the National Ambient Air Quality Standards of India, NO2 limit must be 40. So let's check in how many locations in each state does the N02 pollutant value is greater than 40.

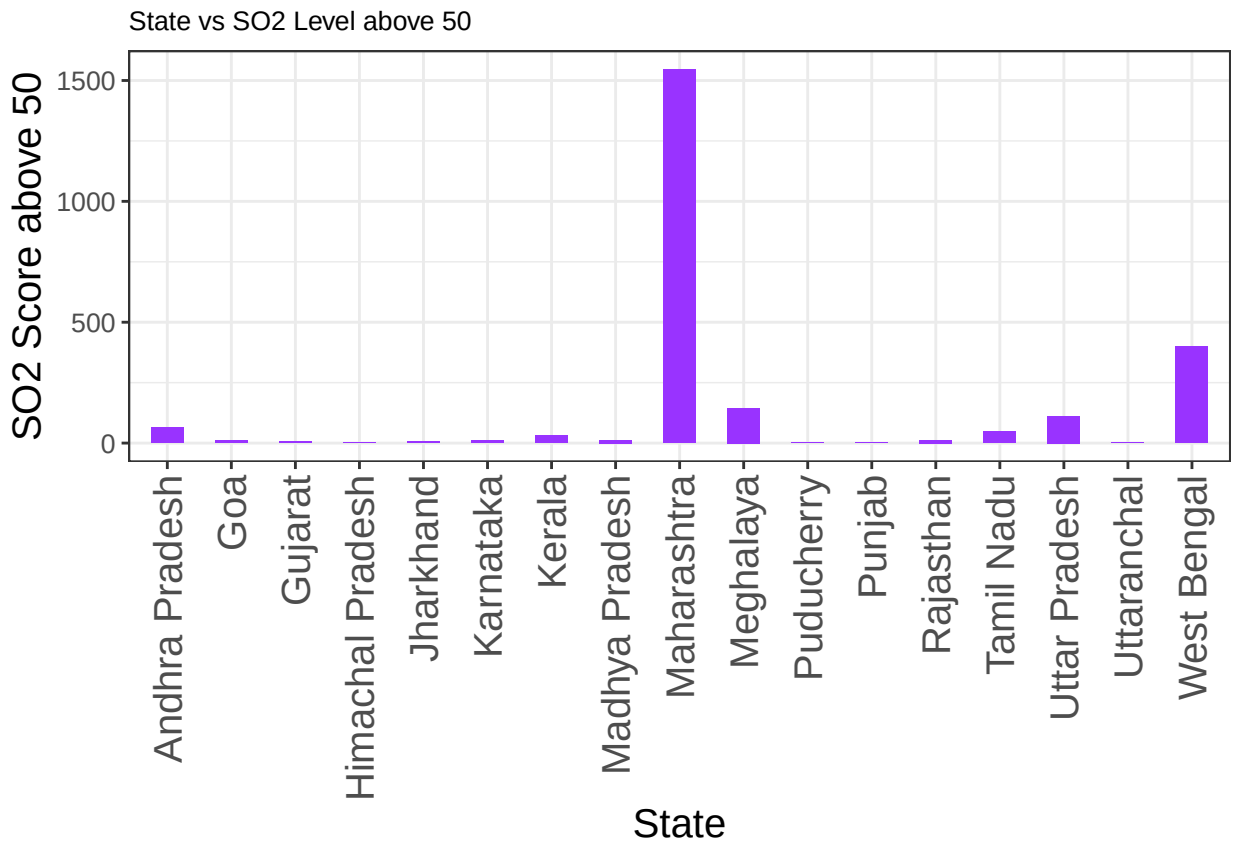
```
ggplot(pollution_data[pollution_data$no2>40,], aes(state))+
  geom_bar(stat="count", width = 0.5, fill="darkblue")+
  labs(x="State",
       y="NO2 Score above 40",
       title="State vs NO2 level above 40 ")+
  theme_bw()+
  theme(plot.title = element_text(size=10),axis.text.x= element_text(size=15),
        axis.text.y= element_text(size=10), axis.title=element_text(size=15)) +
 theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



Surprisingly, West Bengal state tops the list which is followed by Maharashtra. And Delhi and Uttar Pradesh have considerably small count of locations with NO2 levels more than the permissable limit.

Similarly, let's check for the SO2 pollutant. According to the National Ambient Air Quality Standards of India, SO2 limit must be 50. So let's check in how many locations in each state does the S02 pollutant value is greater than 50.

```
ggplot(pollution_data[pollution_data$so2>50,], aes(state))+
  geom_bar(stat="count", width = 0.5, fill="#9933FF")+
  labs(x="State",
       y="SO2 Score above 50",
       title="State vs SO2 Level above 50 ")+
  theme_bw()+
  theme(plot.title = element_text(size=10),axis.text.x= element_text(size=15),
        axis.text.y= element_text(size=10), axis.title=element_text(size=15)) +
 theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



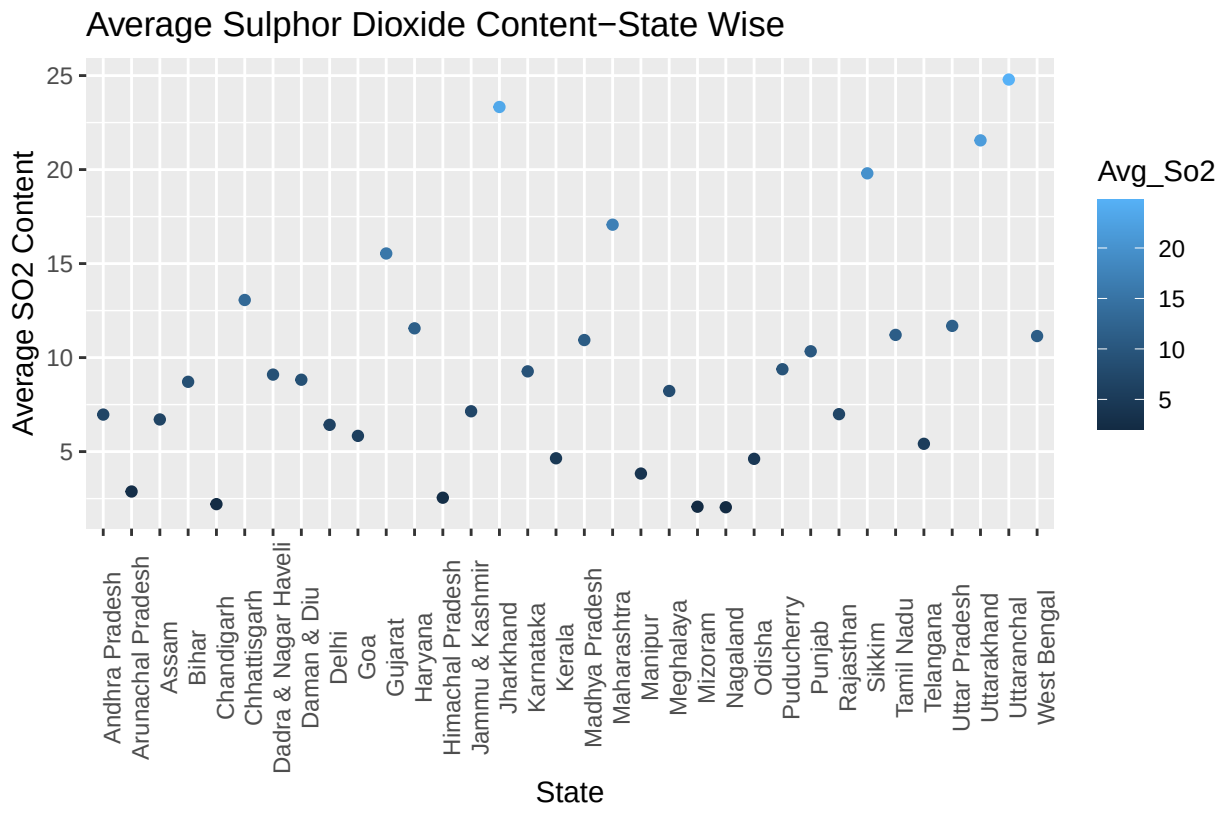
Interesting to find out that, Maharashtra state is followed by West Bengal state in the list containing locations more than the permissable limit of 50

Okay, now in that case, it is very anxious to find out the average sulphur dioxide content that is being emitted into the air in each state. Let's find out if Maharashtra or West Bengal tops the list again.

```
by_state_wise <-pollution_data%>%group_by(state)%>%summarise(Avg_So2=mean(so2,na.rm=TRUE), Avg_No2=mean


ggplot(by_state_wise,aes(x=state,y=Avg_So2,color=Avg_So2)) +
  geom_point(stat="identity") +
  theme(axis.text.x =element_text(angle=90)) +
```

```
  ggtitle("Average Sulphor Dioxide Content-State Wise") +
  xlab(label="State") +
  ylab(label="Average SO2 Content") +
  labs(caption="")
```



Throwing away all the assumptions made before, Jharkand state emits the highest Average Sulphur Dioxide content into air.

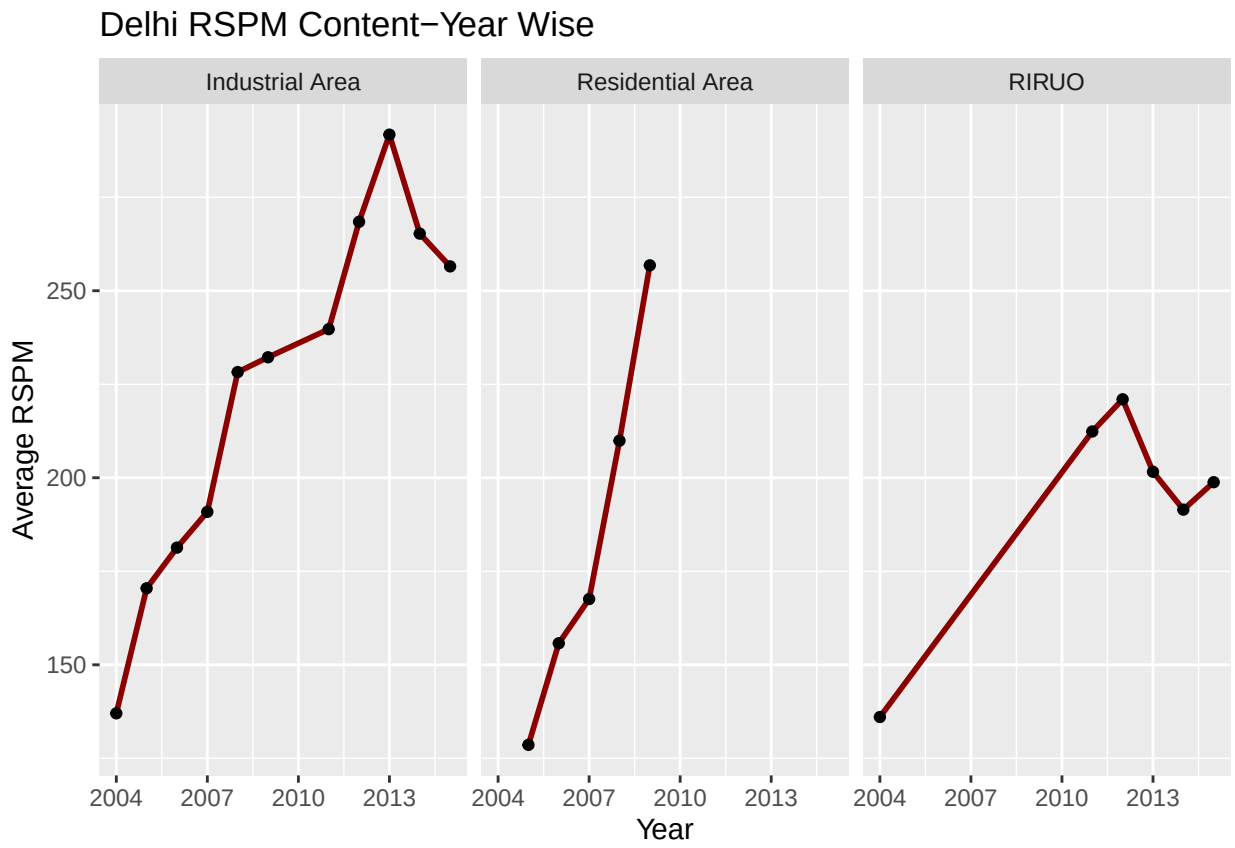
Now let's have a look at Delhi state which is actually a Union Territory & the capital of India and also one of the top polluted city in India. Let's chcck how the particule matter more than 10 PPM (i.e., RSPM value here) has been varying from 2004 - 2014

```
Delhi <-pollution_data%>%filter(state=="Delhi")%>%group_by(Year,type)%>%summarise(Avg_So2=mean(so2,na.rm
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
ggplot(Delhi,aes(x=Year,y=Avg_Rspm)) +
  geom_line(size=1,color="darkred") +
  geom_point()+
  facet_wrap(~type) +
  ggtitle("Delhi RSPM Content-Year Wise")+
  xlab("Year") +
  ylab("Average RSPM")
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
```



It is clearly evident that the RSPM value increased a lot in both Industrial and Residential area over the period of 2004 - 2014 which contributed a lot for the pollution in Delhi.

Data Analysis

- Respirable suspended particulate matter (RSPM), the deadliest pollutant in the city air.
- Let's analyze which state is worst in case of RSPM

```
state_rspm <- pollution_data %>% select(state,rspm) %>% group_by(state) %>% summarize(rspm_mean = mean(r


state_rspm[order(-state_rspm$rspm_mean),]
```

```
## # A tibble: 34 x 2
##    state         rspm_mean
##    <chr>             <dbl>
##  1 Delhi              200.
##  2 Uttar Pradesh      177.
##  3 Jharkhand          169.
##  4 Punjab             166.
##  5 Uttarakhand        149.
##  6 Haryana            148.
```

```
##  7 Rajasthan          142.
##  8 Chhattisgarh       124.
##  9 Bihar              124.
## 10 Uttaranchal        123.
## # ... with 24 more rows
```

We found that Delhi has the worst average RSPM among the states

Now let's check the same among the cities

```
location_rspm <- pollution_data %>% select(location,rspm) %>% group_by(location) %>% summarize(rspm_mean


location_rspm[order(-location_rspm$rspm_mean),]
```

```
## # A tibble: 284 x 2
##    location       rspm_mean
##    <chr>              <dbl>
##  1 Ghaziabad           250.
##  2 West Singhbhum      246.
##  3 Bareilly            233.
##  4 Allahabad           231.
##  5 Ludhiana            218.
##  6 Jharia              211.
##  7 Raipur              209.
##  8 Khanna              208.
##  9 Mathura             206.
## 10 Gwalior             200.
## # ... with 274 more rows
```

Seems to be, Ghaziabad has the worst average RSPM value among all the cities.

Now, it would be great to found which type of area is more polluted.

```
type_rspm <- pollution_data %>% select(type,rspm) %>% group_by(type) %>% summarize(rspm_mean = mean(rspm


type_rspm[order(-type_rspm$rspm_mean),]
```

```
## # A tibble: 4 x 2
##   type             rspm_mean
##   <chr>                <dbl>
## 1 Industrial Area      122.
## 2 Residential Area     103.
## 3 RIRUO                102.
## 4 Sensitive Area        99.0
```

As expected, the Industrial area is the most polluted one.

Conclusion & Biases

We can infer from the above data analysis that as more number of industries are located in the city of Ghaziabad which is part of Delhi state, it is one of the top polluted city because of presence of high amount

of pollutants like RSPM. In addition, we can see West Singhbhum which is famous for Chromites deposits and iron ore mine. So we can observe more amount of RSPM in air along with highest average sulphur dioxide content (SO2) in air. And also the top 5 polluted cities observed in our analysis are more or less from the top 3 polluted states i.e., Delhi, Uttar Pradesh, Jharkhand.

Most of the industries in the above mentioned areas are the main reasons for pollutions and they seems to be least bothered about the air quality and are least concerned about the environmental policy. And even the residential area in Delhi contributes a lot to the pollution in Delhi state. So industrial and residential areas are the possible reasons for more pollution in Northern India. So these areas are dominating the lists with highest levels of pollutants in air.

All these factors/biases are playing a crucial role in polluting the air. These can be reduced by having a strict environmental policy and proper steps to relocate the industries from this area.

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur ... 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] corrplot_0.92   forcats_0.5.2   stringr_1.4.1   dplyr_1.0.10
##  [5] purrr_0.3.5     readr_2.1.3     tidyr_1.2.1     tibble_3.1.8
##  [9] tidyverse_1.3.2 ggplot2_3.4.0
##
## loaded via a namespace (and not attached):
##  [1] lubridate_1.9.0     assertthat_0.2.1    digest_0.6.30
##  [4] utf8_1.2.2          R6_2.5.1            cellranger_1.1.0
##  [7] backports_1.4.1     reprex_2.0.2        evaluate_0.17
## [10] highr_0.9           httr_1.4.4          pillar_1.8.1
## [13] rlang_1.0.6         curl_4.3.3          googlesheets4_1.0.1
## [16] readxl_1.4.1        rstudioapi_0.14     rmarkdown_2.17
## [19] labeling_0.4.2      googledrive_2.0.0   bit_4.0.4
## [22] munsell_0.5.0       broom_1.0.1         compiler_4.2.1
## [25] modelr_0.1.9        xfun_0.34           pkgconfig_2.0.3
## [28] htmltools_0.5.3     tidyselect_1.2.0    fansi_1.0.3
## [31] crayon_1.5.2        tzdb_0.3.0          dbplyr_2.2.1
## [34] withr_2.5.0         grid_4.2.1          jsonlite_1.8.3
## [37] gtable_0.3.1        lifecycle_1.0.3     DBI_1.1.3
## [40] magrittr_2.0.3      scales_1.2.1        cli_3.4.1
## [43] stringi_1.7.8       vroom_1.6.0         farver_2.1.1
## [46] fs_1.5.2            xml2_1.3.3          ellipsis_0.3.2
## [49] generics_0.1.3      vctrs_0.5.0         tools_4.2.1
## [52] bit64_4.0.5         glue_1.6.2          hms_1.1.2
## [55] parallel_4.2.1      fastmap_1.1.0       yaml_2.3.6
```

```
## [58] timechange_0.1.1    colorspace_2.0-3    gargle_1.2.1
## [61] rvest_1.0.3         knitr_1.40          haven_2.5.1
```