Mini Project - Analysis of Airquality And Weather data of Indian Cities

Riyas A R (2023MCS120001)

11/4/2023

OBJECTIVES

The objectives planned for this mini project is as below:

1. Analysis of air quality levels in different cities
2. Analysis to find the relationship between population and air quality by city
3. Analysis to find the relationship between air pollution and specific weather conditions in Delhi
4. Use of Statistical modelling predict AQI variable based on related parameters.

PRELIMINARY ANALYSIS

```
library(statisticalModeling)
```

1. The following libraries are used for analysis:

```
## Loading required package: ggplot2
```

```
library(mosaic)
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, binom.test, cor, cor.test, cov, fivenum, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library(rpart)
library(dplyr)
library(ggplot2)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

```
library(naniar)
library(simputation)
```

```
##
## Attaching package: 'simputation'

## The following object is masked from 'package:naniar':
##
##     impute_median
```

```
library(rpart)
```

```
myWD <- getwd();
#Loading Air Quality Data
stations<-read.csv(paste(myWD,"/Datasets/Air_Quality/stations.csv",sep = ""))
station_day<-read.csv(paste(myWD,"/Datasets/Air_Quality/station_day.csv",sep = ""))


#Loading Weather Data of Delhi
Delhi_Safdarjung<-read.csv(paste(myWD,"/Datasets/Temp_And_Precipitation_Cities_IN/Delhi_NCR_1990_2022_Sa
Station_GeoLocation<-read.csv(paste(myWD,"/Datasets/Temp_And_Precipitation_Cities_IN/Station_GeoLocatio

#Loading population Data set of Indian Cities on 2021
population_india_City <- read.csv(paste(myWD,"/Datasets/Population2021_India_Cities.csv",sep = ""))
```

2. Loaded the Air Quality data and Wether data for processing

```
# Filter the Air Quality data for the years 2015 to 2020
station_day_filtered <- station_day %>%  filter(Date >= 2015 & Date <= 2020)

# Adding column City in Air Quality data by day
station_day_filtered$City <- stations$City[match(station_day_filtered$StationId, stations$StationId)]
```

```r
# Inspect the filtered data
str(station_day_filtered)
```

3. Data Cleaning and Preprocessing

```
## 'data.frame':    88671 obs. of  17 variables:
##  $ StationId : chr  "AP001" "AP001" "AP001" "AP001" ...
##  $ Date      : chr  "2017-11-24" "2017-11-25" "2017-11-26" "2017-11-27" ...
##  $ PM2.5     : num  71.4 81.4 78.3 88.8 64.2 ...
##  $ PM10      : num  116 124 129 135 104 ...
##  $ NO        : num  1.75 1.44 1.26 6.6 2.56 5.23 4.69 4.58 7.71 0.97 ...
##  $ NO2       : num  20.6 20.5 26 30.9 28.1 ...
##  $ NOx       : num  12.4 12.1 14.8 21.8 17 ...
##  $ NH3       : num  12.2 10.7 10.3 12.9 11.4 ...
##  $ CO        : num  0.1 0.12 0.14 0.11 0.09 0.16 0.12 0.1 0.1 0.15 ...
##  $ SO2       : num  10.8 15.2 27 33.6 19 ...
##  $ O3        : num  109 127 117 112 138 ...
##  $ Benzene   : num  0.17 0.2 0.22 0.29 0.17 0.21 0.16 0.17 0.25 0.23 ...
##  $ Toluene   : num  5.92 6.5 7.95 7.63 5.02 4.71 3.52 2.85 2.79 3.82 ...
##  $ Xylene    : num  0.1 0.06 0.08 0.12 0.07 0.08 0.06 0.04 0.07 0.04 ...
##  $ AQI       : num  NA 184 197 198 188 173 165 191 191 227 ...
##  $ AQI_Bucket: chr  "" "Moderate" "Moderate" "Moderate" ...
##  $ City      : chr  "Amaravati" "Amaravati" "Amaravati" "Amaravati" ...
```

```r
# For analysing temperature data, we consider only the Delhi City
# Convert 'time' to Date format
Delhi_Safdarjung$Date <- as.Date(Delhi_Safdarjung$time, format = "%d-%m-%Y")
Temp_Delhi_City_filtered <- Delhi_Safdarjung %>%  filter(Date >= as.Date("2015-01-01") & Date <= as.Dat
str(Temp_Delhi_City_filtered)
```

```
## 'data.frame':    2192 obs. of  6 variables:
##  $ time: chr  "01-01-2015" "02-01-2015" "03-01-2015" "04-01-2015" ...
##  $ tavg: num  14.9 14.7 15 14.2 13.9 12 9.8 10.1 10.5 11.2 ...
##  $ tmin: num  8.8 10.2 NA NA 7.8 NA 6.4 7.2 6.8 7.2 ...
##  $ tmax: num  21.6 21.6 16.8 19 20.7 20.7 17.6 14.5 16.3 18.3 ...
##  $ prcp: num  NA 5.1 7.1 0 NA NA NA NA NA NA ...
##  $ Date: Date, format: "2015-01-01" "2015-01-02" ...
```

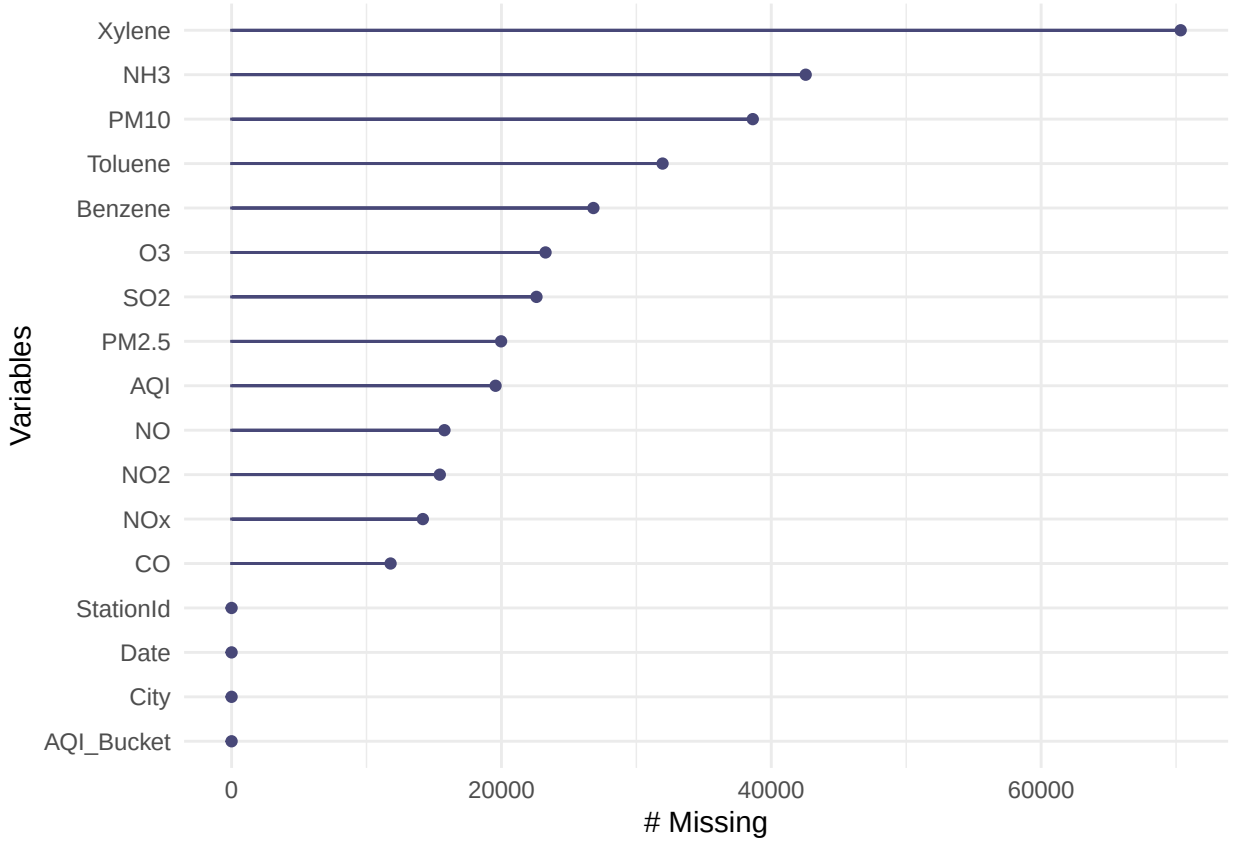
```r
# Summarise missingness in each variable of the `airquality` dataset
miss_var_summary(station_day_filtered)
```

4. Checking the missing Data

```
## # A tibble: 17 x 3
##    variable   n_miss pct_miss
##    <chr>       <int>    <dbl>
##  1 Xylene      70342     79.3
##  2 NH3         42551     48.0
##  3 PM10        38628     43.6
##  4 Toluene     31943     36.0
##  5 Benzene     26816     30.2
##  6 O3          23269     26.2
##  7 SO2         22597     25.5
##  8 PM2.5       19975     22.5
```

```
##  9 AQI          19566     22.1
## 10 NO           15778     17.8
## 11 NO2          15433     17.4
## 12 NOx          14176     16.0
## 13 CO           11783     13.3
## 14 StationId        0      0
## 15 Date             0      0
## 16 AQI_Bucket       0      0
## 17 City             0      0
```

```
# Visualize the number of missings in variables in air quality data using `gg_miss_var()`
gg_miss_var(station_day_filtered)
```




```
# Use `replace_with_na_at()` to replace all missing values with NA
station_day_filtered<-replace_with_na_at(station_day_filtered,
                                         .vars = c("PM2.5", "PM10", "NO", "NO2", "NOx", "NH3", "CO", "S
                                         ~.x %in% c("na", "", "NaN", " "))
#For predicting AQI variable, handle missing data by imputing the mean within each city
station_day_filtered <- station_day_filtered %>%
  group_by(City) %>%
  mutate(
    PM2.5 = ifelse(is.na(PM2.5), mean(PM2.5, na.rm = TRUE), PM2.5),
    PM10 = ifelse(is.na(PM10), mean(PM10, na.rm = TRUE), PM10),
    NO2 = ifelse(is.na(NO2), mean(NO2, na.rm = TRUE), NO2),
      NO = ifelse(is.na(NO), mean(NO, na.rm = TRUE), NO),
    NOx = ifelse(is.na(NOx), mean(NOx, na.rm = TRUE), NOx),
    NH3 = ifelse(is.na(NH3), mean(NH3, na.rm = TRUE), NH3),
    CO = ifelse(is.na(CO), mean(CO, na.rm = TRUE), CO),
```

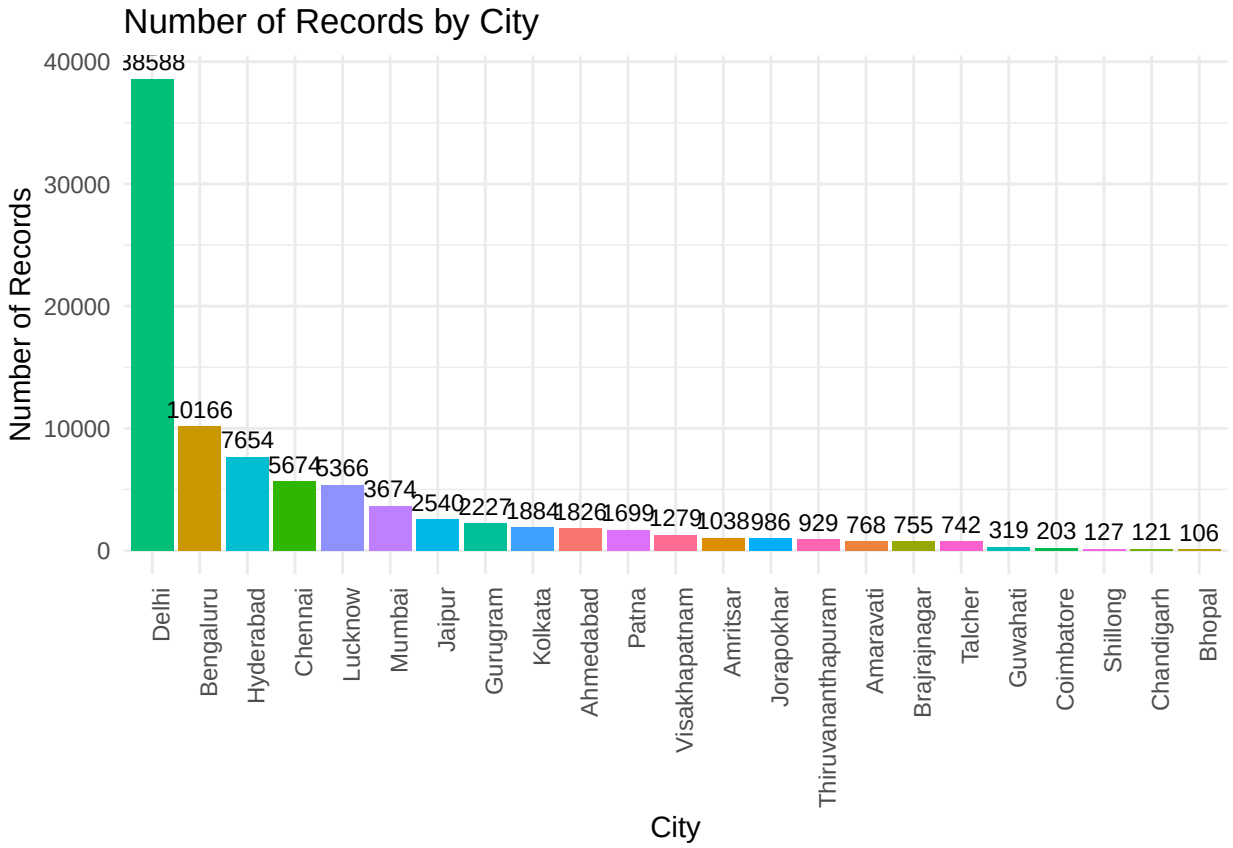
```
    SO2 = ifelse(is.na(SO2), mean(SO2, na.rm = TRUE), SO2),
    O3 = ifelse(is.na(O3), mean(O3, na.rm = TRUE), O3),
      Benzene = ifelse(is.na(Benzene), mean(Benzene, na.rm = TRUE), Benzene),
    Toluene = ifelse(is.na(Toluene), mean(Toluene, na.rm = TRUE), Toluene),
    NO2 = ifelse(is.na(NO2), mean(NO2, na.rm = TRUE), NO2),
      Xylene = ifelse(is.na(Xylene), mean(Xylene, na.rm = TRUE), Xylene),
    AQI = ifelse(is.na(AQI), mean(AQI, na.rm = TRUE), AQI)
  ) %>%
  ungroup()  # Ungroup the data to avoid unintended effects in subsequent operations
```

```
# Filling missing data in AQI filed of Airquality data using Imputation methods

station_day_filtered <- bind_shadow(station_day_filtered) %>%
  impute_lm(PM10 ~ PM2.5 + NO + NO2 + O3  ) %>%
  impute_lm(NH3 ~ PM2.5 + NO + NO2 +  O3  ) %>%
  impute_lm(Xylene ~ PM2.5 + NO + NO2 +  O3  ) %>%
  impute_lm(Toluene ~ PM2.5 + NO + NO2 +  O3  ) %>%
  impute_lm(NOx ~ PM2.5 + NO + NO2 +  O3  ) %>%
  impute_lm(Benzene ~ PM2.5 + NO + NO2 +  O3  ) %>%
  add_label_shadow()
miss_var_summary(station_day_filtered)
```

```
## # A tibble: 35 x 3
##    variable  n_miss pct_miss
##    <chr>      <int>    <dbl>
##  1 StationId      0        0
##  2 Date           0        0
##  3 PM2.5          0        0
##  4 PM10           0        0
##  5 NO             0        0
##  6 NO2            0        0
##  7 NOx            0        0
##  8 NH3            0        0
##  9 CO             0        0
## 10 SO2            0        0
## # i 25 more rows
```

5. Analysis of available records If we are going to compare cities, the comparison must ideally be between cities with comparable size of records. However, some cities have more records while others have far less. Therefore, it was necessary to take this into consideration while building the case.

```
# Calculate the number of records for each city
records_by_city <- station_day_filtered %>%
  group_by(City) %>%  summarize(Records = n()) %>%  arrange(desc(Records))
# Create a bar graph
bar_graph <- ggplot(records_by_city, aes(x = reorder(City, -Records), y = Records, fill = City)) +
  geom_bar(stat = "identity") +
  labs(title = "Number of Records by City", x = "City", y = "Number of Records") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1), legend.position = "none") + # Rotate x-axis
  geom_text(aes(label = Records), vjust = -0.5, size = 3)  # Add data labels

# Display the bar graph
print(bar_graph)
```

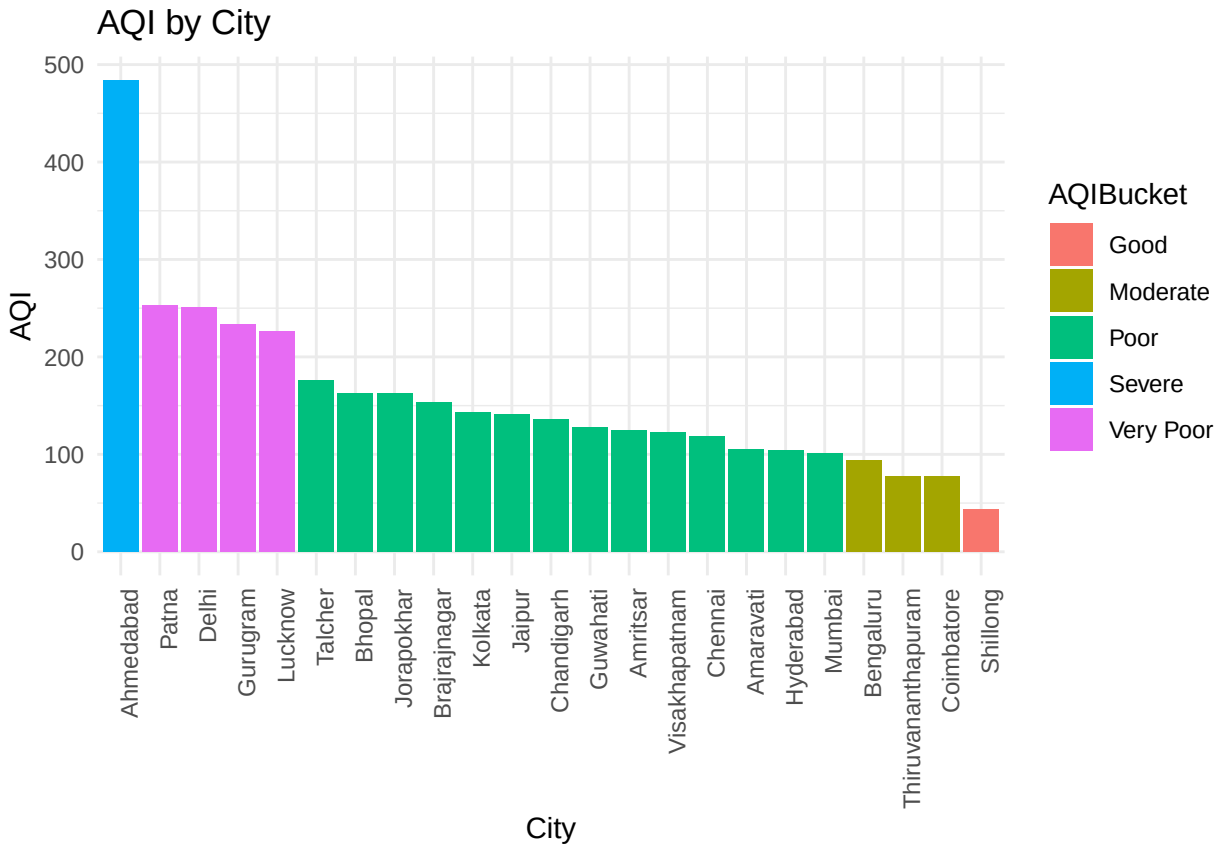


DETAILED ANALYSIS

1. Visually compare the air quality levels in different cities Identify cities with the highest and lowest AQI values. This can help in understanding the variation in air quality across different locations.

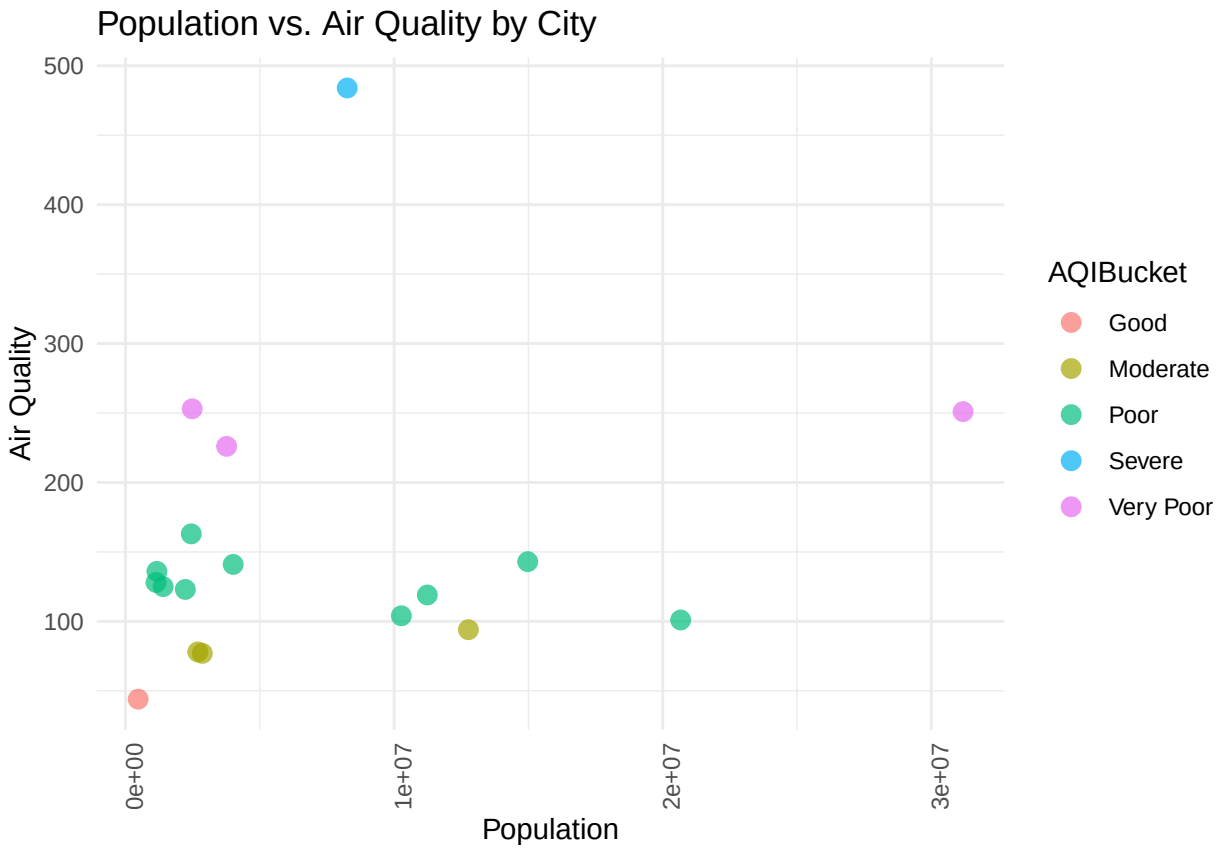
```
# Define a function to calculate AQI bucket with labels
calculate_AQI_bucket <- function(AQI_value) {
  if (AQI_value >= 0 && AQI_value <= 50) {
    return("Good")
  } else if (AQI_value >= 51 && AQI_value <= 100) {
    return("Moderate")
  } else if (AQI_value >= 101 && AQI_value <= 200) {
    return("Poor")
  } else if (AQI_value >= 201 && AQI_value <= 300) {
    return("Very Poor")
  } else {
    return("Severe")
  }
}
# Create a bar chart for AQI by city
station_day_grp_city <- station_day_filtered %>%  group_by(City) %>%
  summarise(avg_AQI = round(mean(AQI, na.rm = TRUE)), AQIBucket = calculate_AQI_bucket(avg_AQI)) %>%
  arrange(desc(avg_AQI))

bar_chart <- ggplot(station_day_grp_city, aes(x = reorder(City, -avg_AQI), y = avg_AQI, fill = AQIBucke
```

```
  geom_bar(stat = "identity") +
  labs(title = "AQI by City", x = "City", y = "AQI") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) # Rotate x-axis labels


# Display the bar chart
bar_chart
```



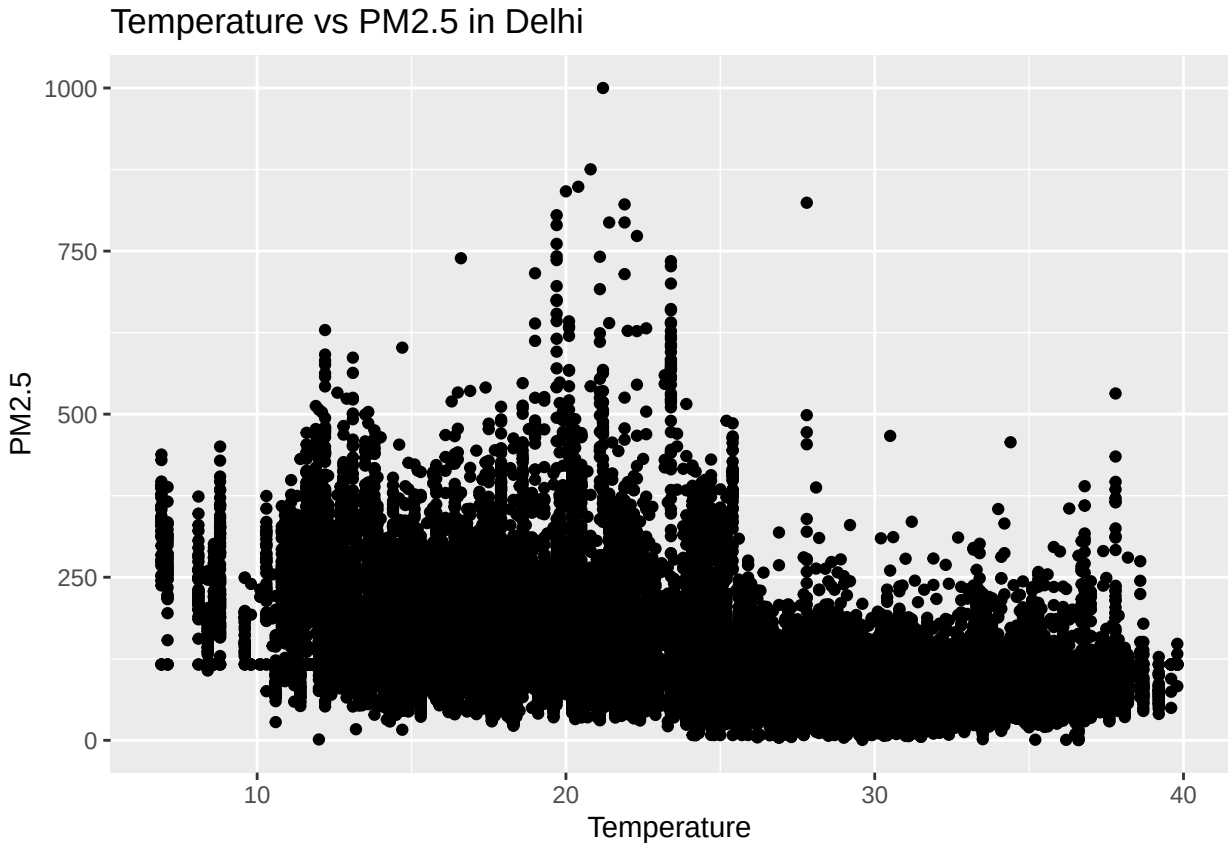
2. Analyzing to find the relationship between population and air quality by city It can provide insights into how population density might be associated with variations in air quality. For this analysis I am using additional data set of Population2021_India_Cities.csv

```
population_india_City <- read.csv(paste(myWD,"/Datasets/Population2021_India_Cities.csv",sep = ""))
merged_quality_population <-left_join(station_day_grp_city, population_india_City, by = "City")
merged_quality_population <- na.omit(merged_quality_population)
merged_quality_population <- merged_quality_population[, -which(names(merged_quality_population) == "X")]
merged_quality_population <- merged_quality_population %>%  arrange(desc(Population.2021.))

ggplot(merged_quality_population, aes(x = Population.2021., y = avg_AQI, color = AQIBucket)) +
  geom_point(size = 3, alpha = 0.7) +
  labs(x = "Population", y = "Air Quality") +
  ggtitle("Population vs. Air Quality by City") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



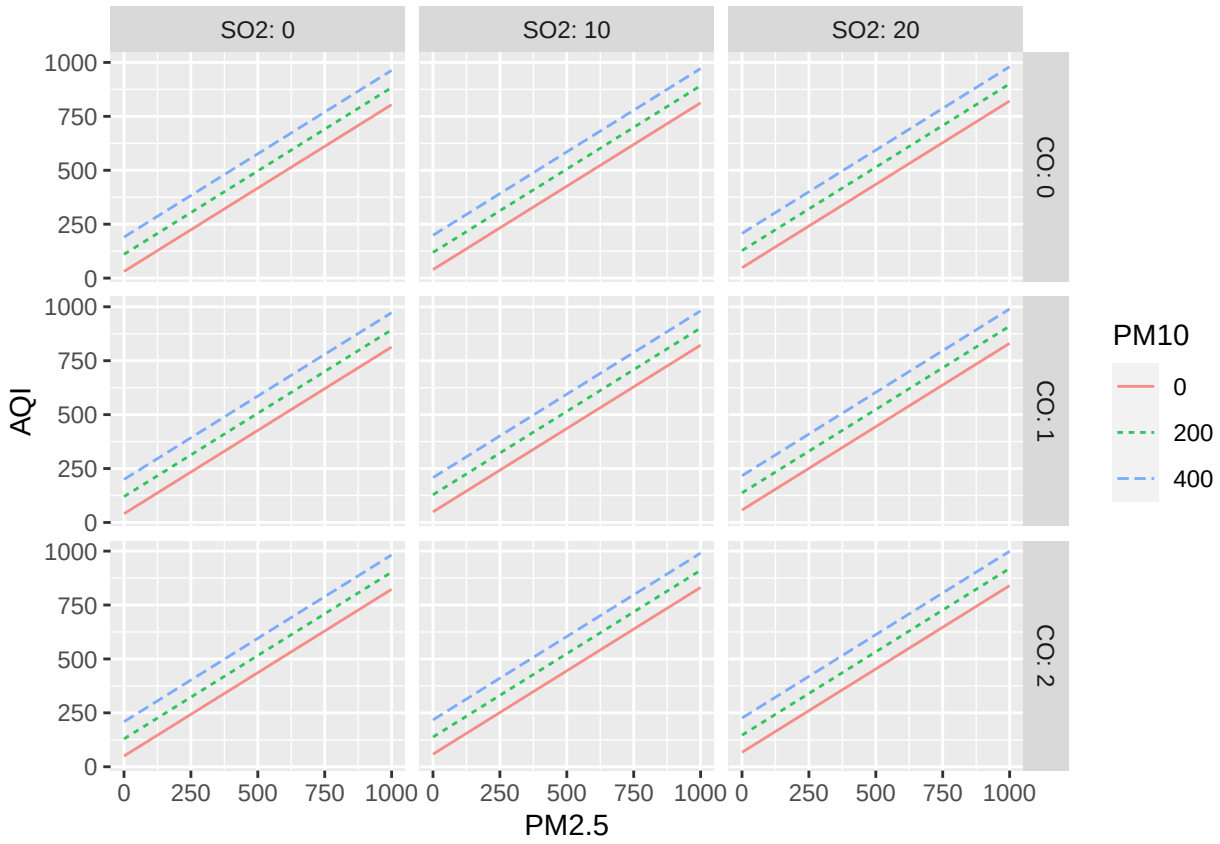
3. Analyzing to find the relationship between air pollution and specific weather conditions in Delhi City The goal is to explore how air pollution (PM2.5) changes during specific temperature condition in Delhi city

```
AQ_day_filtered<-station_day_filtered
AQ_day_filtered$Date <- as.Date(AQ_day_filtered$Date, format = "%Y-%m-%d")
Temp_pm25_delhi <- left_join(Temp_Delhi_City_filtered,
                         AQ_day_filtered %>% filter(City == "Delhi") %>% select(Date, PM2.5),
                         by = "Date")
Temp_pm25_delhi <- Temp_pm25_delhi %>% filter(!is.na(PM2.5))

# Scatter plot to visualize the relationship between temperature and pm25
ggplot(Temp_pm25_delhi, aes(x = tavg, y = PM2.5)) +
  geom_point() +
  labs(title = "Temperature vs PM2.5 in Delhi",
       x = "Temperature",
       y = "PM2.5")
```



4. Predicting the Air Quality Index (AQI) using linear and recursive partitioning models.
Prediction of AQI variable based on parameters PM2.5, PM10, CO, SO2, NO2, O3.

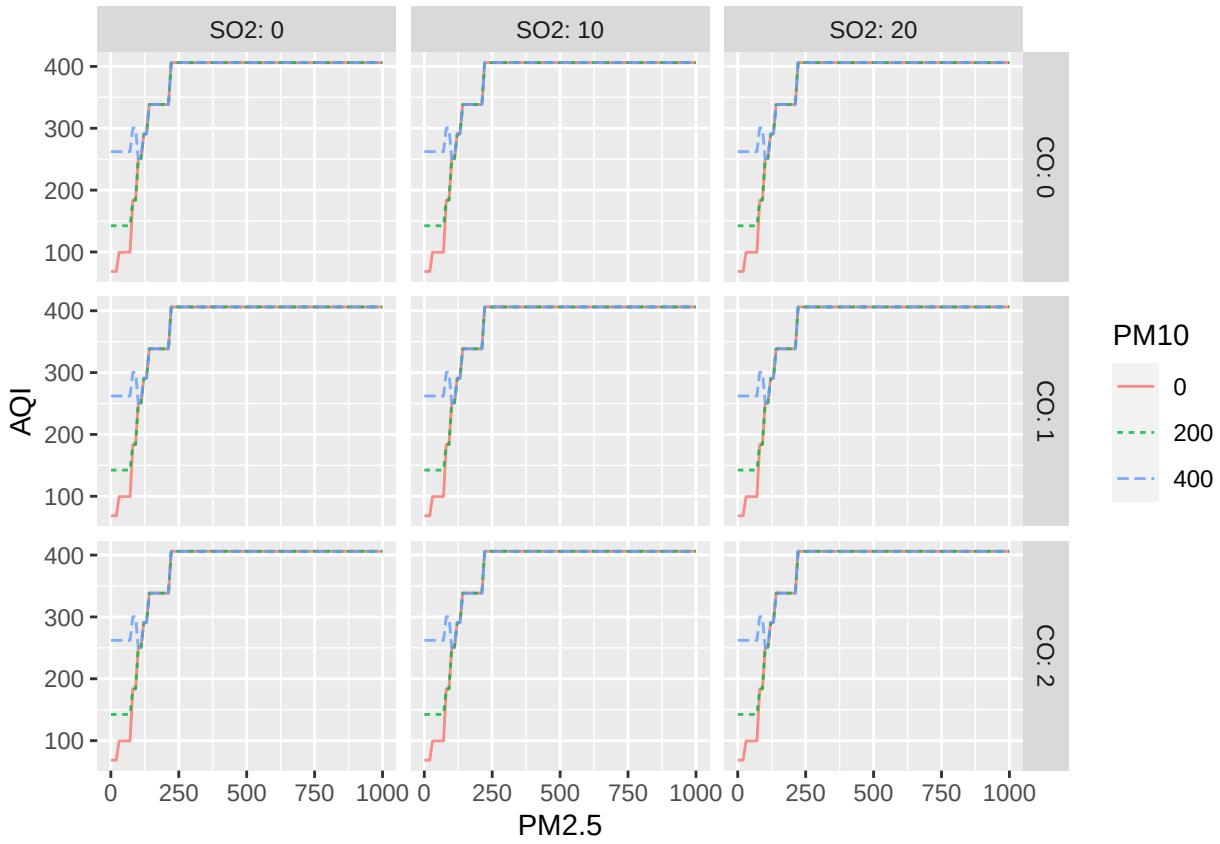
```
# Add a column named training_cases to Air Quality data set(station_day_filtered) consisting of random
# The TRUEs will be the training cases and the FALSEs will be the testing cases.
station_day_filtered$training_cases<-rnorm(nrow(station_day_filtered)) > 0

#Build the linear model AQI~M2.5 + PM10 + CO + SO2 + NO2 + O3 with training cases
linear_model <- lm(AQI~PM2.5 + PM10 + CO + SO2 + NO2 + O3 , data=subset(station_day_filtered,training_ca

#Evaluate the model for the testing cases.
linear_predicts <- evaluate_model(linear_model, data=subset(station_day_filtered,!training_cases))

# Visualize the model
fmodel(linear_model)
```




```
testing_data = subset(station_day_filtered,!training_cases)

# Make predictions on the testing set
predictions <- predict(linear_model, newdata = testing_data)

#Calculate the MSE on the testing data
accuracy <- sqrt(mean((predictions - testing_data$AQI)^2))
print(paste("Root Mean Squared Error (RMSE) of linear modelling :", round(accuracy, 2)))
```

```
## [1] "Root Mean Squared Error (RMSE) of linear modelling : 53.79"
```

```
#Build the recursive partitioning model AQI~PM2.5+PM10+CO+SO2+NO2+O3 with training cases
recursive_model <- rpart(AQI~PM2.5 + PM10 + CO + SO2 + NO2 + O3, data=subset(station_day_filtered,train

#Evaluate the model for the testing cases.
recursive_predicts <- evaluate_model(recursive_model, data=subset(station_day_filtered,!training_cases)

# Visualize the model
fmodel(recursive_model)
```




```
testing_data = subset(station_day_filtered,!training_cases)

# Make predictions on the testing set
predictions <- predict(recursive_model, newdata = testing_data)

#Calculate the MSE on the testing data
accuracy <- sqrt(mean((predictions - testing_data$AQI)^2))
print(paste("Root Mean Squared Error (RMSE) of recursive partitioning model :", round(accuracy, 2)))
```

```
## [1] "Root Mean Squared Error (RMSE) of recursive partitioning model : 51.16"
```

OUTCOMES

1. In Air Quality data set, more records(38588) are available for Delhi city, and less recored(106) are available for Bhopal city.
2. The air quality in Ahmedabad is classified as severe, or extremely unhealthy, based on our comparison of the levels in other cities.
3. Patna, Delhi, Gurugram, and Lucknow have Very Poor (Unhealthy) levels of air quality.
4. The air quality in Shillong City is good.
5. We found that the population and air pollution relationship is not linear and that other factors, such as economic activity, urbanization, energy sources, transportation, land use patterns, weather, and the efficacy of environmental policies, may influence air pollution.
6. The concentration of particulate matter (PM2.5) has been significantly influenced by meteorological parameters especially by temperature. The results show that the concentration of PM2.5 has inverse relationship with the temperature.
7. Using other related parameters in Air Quality data set, predict the AQI values. Here, the AQI value is

predicted using linear modeling under the presumption that the AQI values depend on other air quality parameters.

Results and Discussion

1. In the Air Quality Data Set, missing data is the main challenge.
2. Delhi has the highest number of records (38,588), while Bhopal has the least (106). Variations in the number of records could affect the data's representativeness for some cities, so careful consideration during analysis is necessary.
3. The air quality in Ahmedabad is Severe (Very Unhealthy), while it is Very Poor (Unhealthy) in Patna, Delhi, Gurugram, and Lucknow. To reduce health risks, policy measures and targeted interventions must identify cities with severe air quality.
4. Temperature is one major meteorological factor that affects PM2.5 levels. It is possible to develop efficient pollution reduction strategies by having a clear understanding of these relationships.
5. Predictive modeling allows for forecasting air quality levels, providing a valuable tool for proactive pollution control measures and public health planning.