Air Quality Assessment for the cities in India

Test run on PM2.5 pollutant from 6 Station of Patna city of Bihar

Moorthy M Nair

06/07/2022

The markdown utilises Continuous Ambient Air Quality Monitoring Stations (CAAQMS) information from central server of Central Pollution Control Board (CPCB), India to analyse the efficacy of city specific ground implementation measures on Air Quality (AQ)through robust air quality data analysis. It is envisaged that the analysis shall be instrumental for the decision makers at city level in scaling up the implementation measures.

Understanding the strategy applied in analyzing the AQ dataset.

1. Dataset (24 Hrs average) are downloaded from (https://app.cpcbccr.com/ccr/#/caaqm-dashboard-all/caaqm-landing)
2. Analysis are limited to PM_{10} and $PM_{2.5}$
3. Analysis is carried out with respect to Financial Year (April to March)
4. A Minimum of 75% data requirement in each quarter and FY is considered as pre-requisites
5. In between missing values are interpolated using Linear Interpolation method
6. In case of missing value at the start date of the range, A average of first 30 days shall be used to replace the missing value. In case of missing value at the end date of the range, average of last 30 days shall be used as its replacement
7. Data missing consecutively for more than 30 days were eliminated for the assessment
8. Outliers were assessed on Quarterly basis (2*S.D > Mean Value - Daily Value > -2*S.D)
9. Air Quality Index (AQI) <= 250 and AQI <=90 is considered as good days for PM_{10} and $PM_{2.5}$ sub index respectively. However, their might be mismatch in total number of good days when compared against both the sub index and this uncertanity shall be considered judiciously in decision making process.

Mandatory user inputs for the Markdown

1. Chunk 1: Initiate the Working Directory . Users shall input the following
   a) Path: The path where CAAQMS retrieved data are stored
   b) Pollutants that you wish to analyse (PM_{10} or $PM_{2.5}$; One at a time)
   c) Start and End date of the FY for the analysis
2. Chunk 10: Initiate the FY . User shall assign the respective FY accordingly in case of any changes from those mentioned in the chunk.

Note: It is highly suggested to run the chunk of code one at a time for better understanding of the assessment.

Github link for the markdown: https://github.com/moorthynair/Air-Quality-Assessment

In case of any issues identified please feel free to write to moorthymnair@yahoo.in

```r
path = "C:/Users/USER/Downloads/Patna" ##Input path of the data
pollutants = 'PM2.5' ## Criteria pollutant to be analysed
Startdate = as.Date('2019-04-01') ## Start date of analysis
Enddate = as.Date('2023-03-31') ## End date of analysis
```

At the outset, Lets read the essential libraries.

```r
##Ensure libraries are installed prior to running this chunk of code
library(rmarkdown)
library(readxl)
library(dplyr)
library(tidyverse)
library(ggplot2)
library(lubridate)
library(tidyr)
library(zoo)
library(ggrepel)
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=40),tidy=TRUE)
```

Let's check for 3 Mandatory columns prior to initiating the analysis (Date, PM2.5 and PM10). In case of a

```r
collection %>%
    unlist() %>%
    kable()
```

x
Samanpura, Patna - BSPCB has a total columns of number: 3-[From Date,PM2.5,PM10]
Rajbansi Nagar, Patna - BSPCB has a total columns of number: 3-[From Date,PM2.5,PM10]
Muradpur, Patna - BSPCB has a total columns of number: 3-[From Date,PM2.5,PM10]
IGSC Planetarium Complex, Patna - BSPCB has a total columns of number: 3-[From Date,PM2.5,PM10]
Govt. High School Shikarpur, Patna - BSPCB has a total columns of number: 3-[From Date,PM2.5,PM10]
DRM Office Danapur, Patna - BSPCB has a total columns of number: 3-[From Date,PM2.5,PM10]

Let us know first the names of the CAAQMS that are used in the Analysis.

```r
stations %>%
    unlist() %>%
    data.frame(row.names = seq(1:length(stations))) %>%
    rename(Stations = 1) %>%
    kable()
```

Stations
DRM Office Danapur, Patna - BSPCB

Stations
Govt. High School Shikarpur, Patna - BSPCB
IGSC Planetarium Complex, Patna - BSPCB
Muradpur, Patna - BSPCB
Rajbansi Nagar, Patna - BSPCB
Samanpura, Patna - BSPCB

Analyzing the pollution concentration of the city.

Let’s have a look at the summary of the pollutant to be analysed for a city over a certain period as desired.

```
AQ_data = AQ_data %>%
    rowwise(Date) %>%
    summarise(Mean_PM = mean(c_across(everything()),
        na.rm = TRUE)) %>%
    mutate(Mean_PM = round(Mean_PM, digits = 2))

summary(AQ_data$Mean_PM)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA’s
##   11.44   34.55   59.29   76.34  103.85  353.24       1
```

**Very important step

Assign the respective Financial Years (FY 2019-20, FY 2020-21 & FY 2021-22) as required for the analysis.

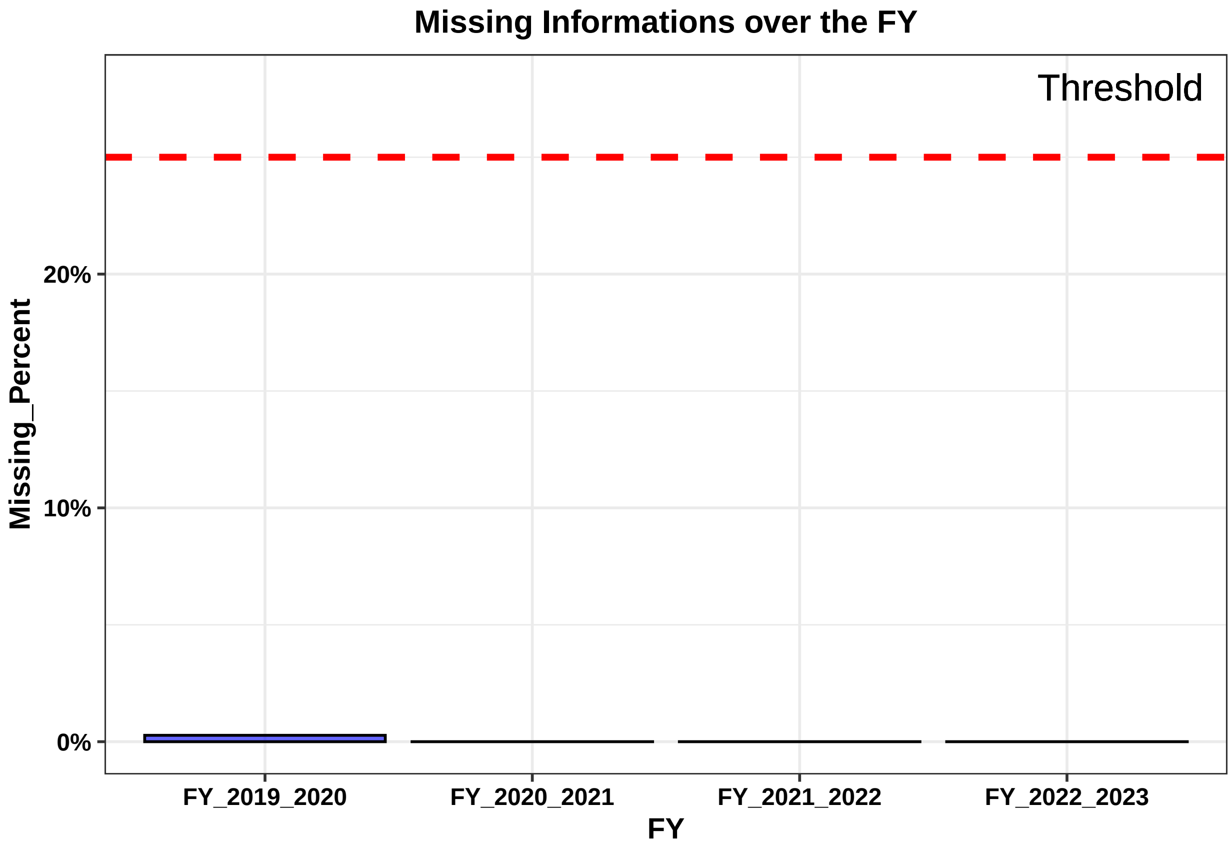
In case of including additional FY, changes must be made in the code below.

```
AQ_data = AQ_data %>%
    mutate(FY = case_when(Date >= as.Date("2018-04-01") &
        Date <= as.Date("2019-03-31") ~ "FY_2018_2019",
        Date >= as.Date("2019-04-01") & Date <=
            as.Date("2020-03-31") ~ "FY_2019_2020",
        Date >= as.Date("2020-04-01") & Date <=
            as.Date("2021-03-31") ~ "FY_2020_2021",
        Date >= as.Date("2021-04-01") & Date <=
            as.Date("2022-03-31") ~ "FY_2021_2022",
        TRUE ~ "FY_2022_2023"))
AQ_data_FY = AQ_data
```

Let us now have check at the missing value for each FY years. We have assumed a Maximum of 25% as thr

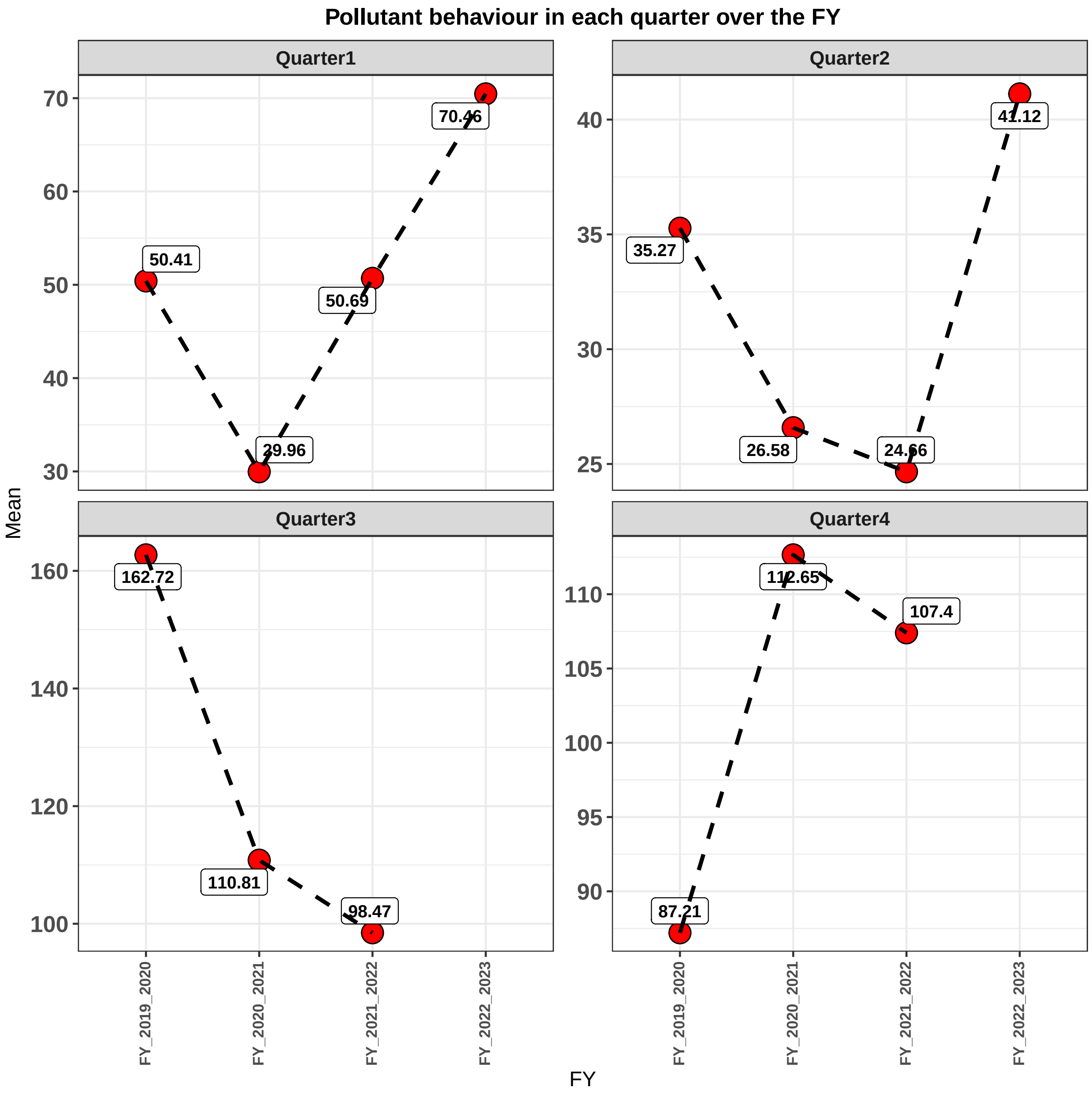
```
## Develop the missing information
## dataframe by performing essential
## mathematical calculations
miss_data = AQ_data %>%
    group_by(FY) %>%
    summarise(Missing_Percent = sum(is.na(Mean_PM) *
        100/n()), total_days_with_values = sum(!is.na(Mean_PM)),
        total_days = n()) %>%
    mutate_if(is.character, as.factor)

## Plotting the graph
```

```
ggplot(miss_data %>%
    mutate_at(vars(Missing_Percent), ~.x/100),
    aes(x = FY, y = Missing_Percent)) + geom_col(col = "black",
    fill = "blue", alpha = 0.6) + geom_hline(aes(yintercept = 0.25),
    lty = 2, col = "red", lwd = 1.2) + scale_y_continuous(labels = scales::label_percent()) +
    ggtitle(label = "Missing Informations over the FY") +
    geom_text(aes(label = "Threshold", y = 0.28,
        x = 4.2), size = 5) + theme_bw() +
    theme(axis.text = element_text(face = "bold",
        color = "black"), axis.title = element_text(face = "bold",
        color = "black"), plot.title = element_text(hjust = 0.5,
        face = "bold", size = 12, color = "black"))
```



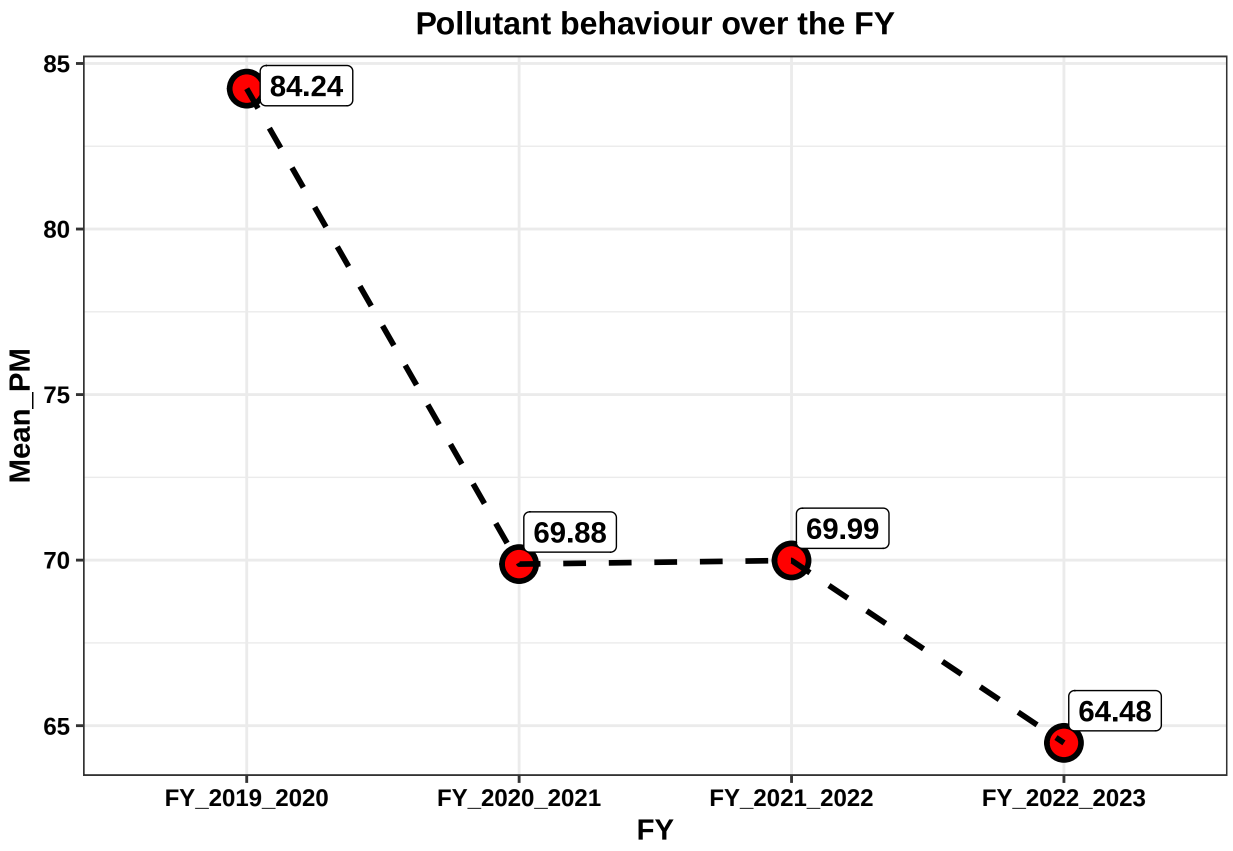
Let's remove the outliers and plot the graph for the respective quarters of FY for further interpretation.

```
AQ_data %>%
    filter(Outlier == "No") %>%
    mutate(Quarter = str_sub(Quarter_year,
        start = 1, end = 8), Mean = mean(Mean_PM,
        na.rm = TRUE), Mean = round(Mean,
        digits = 2)) %>%
    ungroup() %>%
    select(3, 5, 6, 9) %>%
    unique() %>%
    ggplot(aes(x = FY, y = Mean)) + geom_point(shape = 21,
    size = 5, fill = "red") + geom_label_repel(aes(label = Mean),
    size = 3, fontface = "bold") + geom_line(aes(group = 1),
    lty = 2, lwd = 1) + facet_wrap(~Quarter,
    scales = "free_y") + ggtitle(label = "Pollutant behaviour in each quarter over the FY") +
    theme_bw() + theme(axis.text.x = element_text(face = "bold",
    size = 8, angle = 90, vjust = 0.4), axis.text.y = element_text(face = "bold",
    size = 12), strip.text = element_text(face = "bold",
    size = 10), plot.title = element_text(hjust = 0.5,
    face = "bold", size = 12, color = "black"))
```



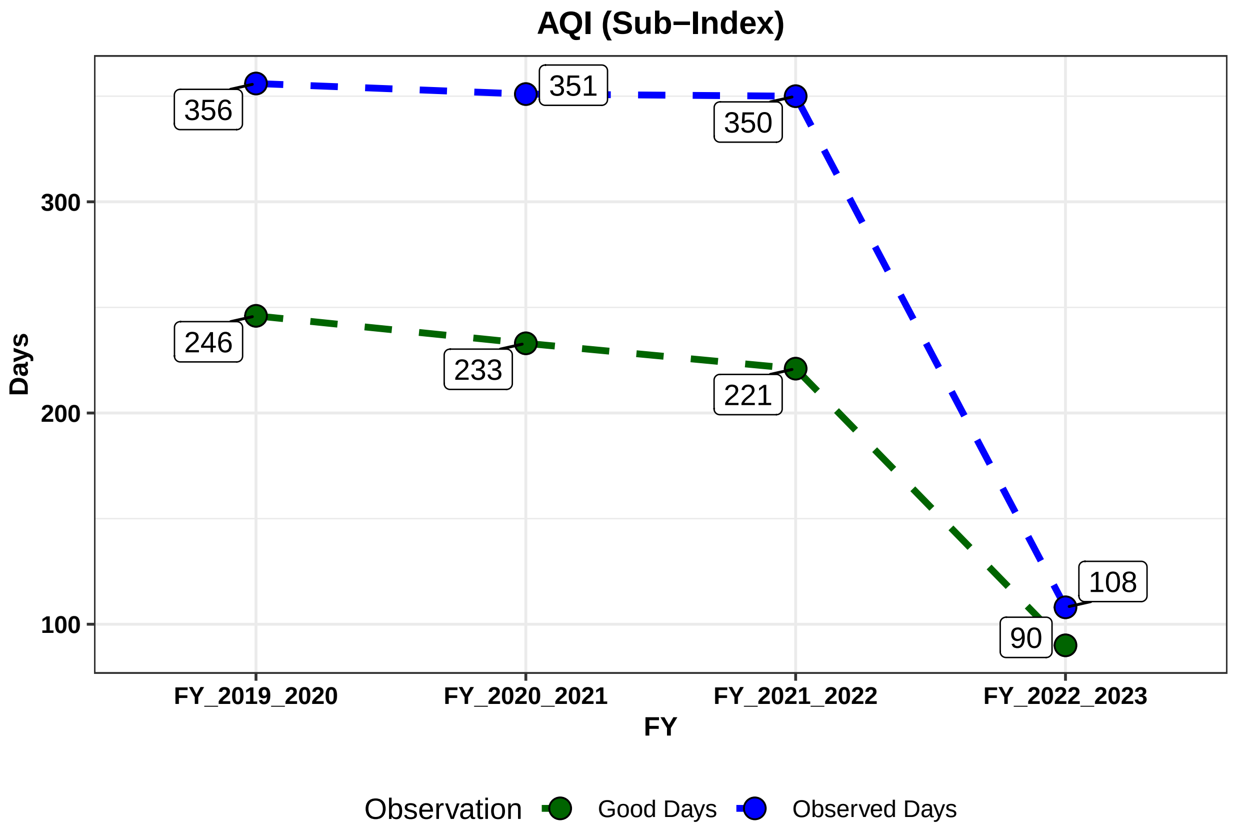
Let us find the overall PM trend over the subsequent FY. (Note Data unavailable for greater than 25%of F

```
## Developing dataframe by performing
## necessary mathematical observations
obs = AQ_data %>%
    group_by(FY) %>%
    summarise(Mean_PM = mean(Mean_PM, na.rm = TRUE),
        Total_observation = n()) %>%
    mutate(Mean_PM = round(Mean_PM, 2)) %>%
    left_join(AQ_data_FY %>%
        group_by(FY) %>%
        count(name = "Total_Days"), by = "FY") %>%
    mutate(Data_availability_percent = Total_observation *
        100/Total_Days, Data_availability_percent = round(Data_availability_percent)) %>%
    select(1, 2, 3, 5) %>%
    mutate_if(is.character, as.factor)


## Plotting the graph
ggplot(obs %>%
    filter(Data_availability_percent > 75),
    aes(x = FY, y = Mean_PM)) + geom_point(fill = "red",
    size = 5, shape = 21, stroke = 1.5) +
    geom_line(aes(group = 1), lwd = 1, lty = 2) +
    geom_label_repel(aes(label = Mean_PM,
        fontface = "bold")) + ggtitle(label = "Pollutant behaviour over the FY") +
    theme_bw() + theme(axis.text = element_text(face = "bold",
    color = "black"), axis.title = element_text(face = "bold",
    color = "black"), plot.title = element_text(hjust = 0.5,
    face = "bold", size = 12, color = "black"))
```



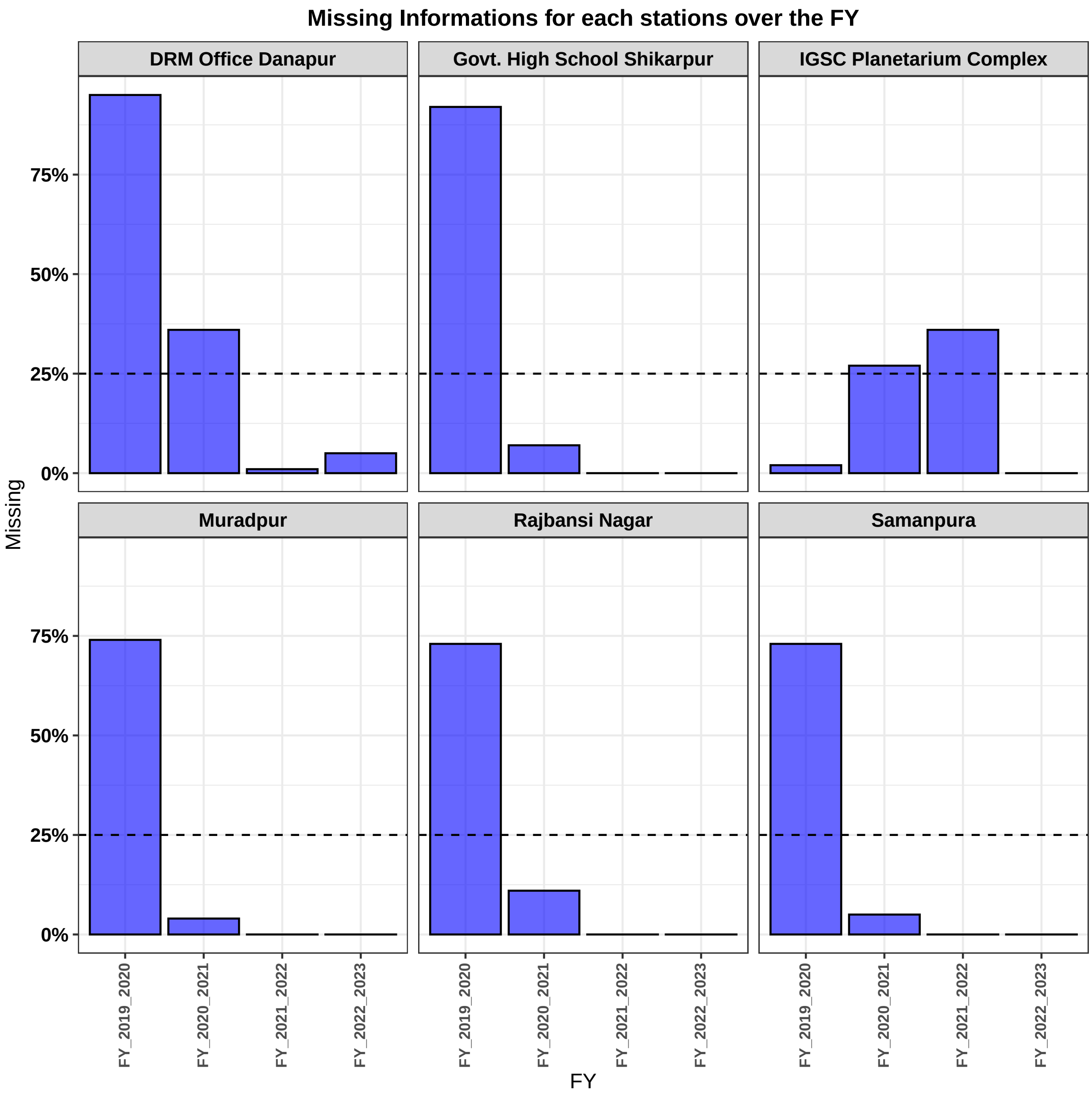
Let us now plot the AQI.

```
if (pollutants == "PM10") {
    AQ_data %>%
        filter(Mean_PM <= 250 & Outlier ==
            "No") %>%
        group_by(FY) %>%
        count(name = "Good Days") %>%
        left_join(AQ_data %>%
            filter(Outlier == "No") %>%
            group_by(FY) %>%
            count(name = "Observed Days"),
            by = "FY") %>%
        pivot_longer(cols = 2:3, names_to = "Observation",
            values_to = "Days") %>%
        mutate_if(is.character, as.factor) %>%
        ggplot(aes(x = FY, y = Days, color = Observation,
            group = Observation)) + geom_line(lwd = 1.2,
        lty = 2) + geom_point(size = 3.5,
        shape = 21, color = "black", aes(fill = Observation)) +
        theme_bw() + geom_label_repel(aes(label = Days),
        color = "black") + ggtitle("AQI (Sub-Index)") +
        theme(axis.text = element_text(face = "bold",
            size = 9, color = "black"), axis.title = element_text(face = "bold",
            size = 10, color = "black"),
            legend.position = "bottom", plot.title = element_text(hjust = 0.5,
                face = "bold", size = 12,
                color = "black")) + scale_color_manual(values = c("darkgreen",
        "blue")) + scale_fill_manual(values = c("darkgreen",
        "blue"))
} else {
    AQ_data %>%
        filter(Mean_PM <= 90 & Outlier ==
            "No") %>%
        group_by(FY) %>%
        count(name = "Good Days") %>%
        left_join(AQ_data %>%
            filter(Outlier == "No") %>%
            group_by(FY) %>%
            count(name = "Observed Days"),
            by = "FY") %>%
        pivot_longer(cols = 2:3, names_to = "Observation",
            values_to = "Days") %>%
        mutate_if(is.character, as.factor) %>%
        ggplot(aes(x = FY, y = Days, color = Observation,
            group = Observation)) + geom_line(lwd = 1.2,
        lty = 2) + geom_point(size = 3.5,
        shape = 21, color = "black", aes(fill = Observation)) +
        theme_bw() + geom_label_repel(aes(label = Days),
        color = "black") + ggtitle("AQI (Sub-Index)") +
        theme(axis.text = element_text(face = "bold",
            size = 9, color = "black"), axis.title = element_text(face = "bold",
            size = 10, color = "black"),
            legend.position = "bottom", plot.title = element_text(hjust = 0.5,
                    face = "bold", size = 12,
                    color = "black")) + scale_color_manual(values = c("darkgreen",
          "blue")) + scale_fill_manual(values = c("darkgreen",
          "blue"))
}
```



Let us now analyze the performance of individual stations.

```
## Calculating the missing information
miss_data = AQ_data_copy %>%
    left_join(AQ_data_FY %>%
        select(1, 3), by = "Date") %>%
    group_by(FY) %>%
    summarise_at(vars(-c(Date)), funs(sum(is.na(.)) *
        100/n()))

## Cleaning the station names
colnames(miss_data) = c("FY", str_split_fixed(stations,
    pattern = ",", n = 2)[1:length(stations)])

## Plotting the missing information
## details for each stations
miss_data %>%
    pivot_longer(cols = -c(FY), names_to = "Stations",
        values_to = "Missing") %>%
    mutate_at(vars(Missing), ~round(./100,
        digits = 2)) %>%
    mutate_at(vars(Stations), ~str_split(.,
        pattern = "_")) %>%
    unnest(cols = c(Stations)) %>%
    mutate(status = if_else(Missing == 1,
        "No Station", NULL)) %>%
    ggplot(aes(x = FY, y = Missing)) + geom_col(colour = "black",
    fill = "blue", alpha = 0.6) + facet_wrap(~Stations) +
    scale_y_continuous(labels = scales::label_percent()) +
    ggtitle("Missing Informations for each stations over the FY") +
    theme_bw() + theme(axis.text.x = element_text(face = "bold",
    size = 8, angle = 90, vjust = 0.4), axis.text.y = element_text(face = "bold",
    size = 10, color = "black"), strip.text = element_text(face = "bold",
    size = 10, color = "black"), plot.title = element_text(hjust = 0.5,
    face = "bold", size = 12, color = "black")) +
    geom_hline(yintercept = 0.25, lty = 2) +
    geom_text(aes(label = status), angle = 90,
        fontface = "bold", hjust = 1.2, colour = "white",
        size = 4.5)
```



Lets have some insights in to the operational period for individual stations. Stations with Minimum 75% da

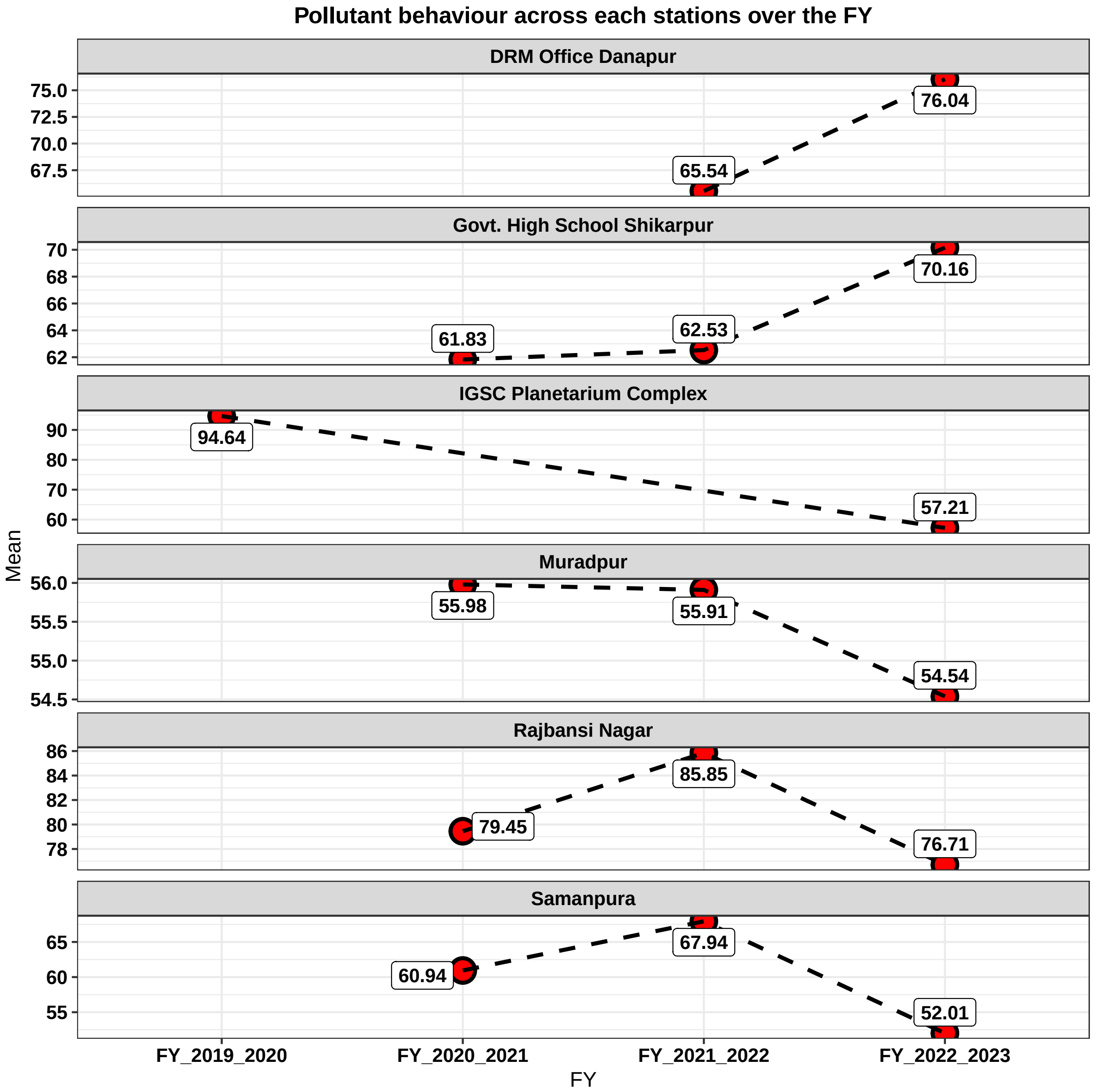
```
## Assign '1' for Missing information
## less than 25% and '0' otherwise
station_rel = ifelse(miss_data[, -c(1)] >
    25, 0, 1)

## Performing row sum to calculate the
## total '1's
station_details = data.frame(apply(station_rel,
    1, sum))

## Binding all the details to a
## dataframe
```

```r
p = data.frame(cbind(miss_data$FY, station_rel)) %>%
    rename(FY = 1) %>%
    gather(key = "Stations", value = "vals",
        -FY) %>%
    filter(vals == 1) %>%
    mutate_at(vars(Stations), ~str_split(.,
        pattern = "_")) %>%
    unnest(cols = c(Stations)) %>%
    group_by(FY) %>%
    mutate(Station = paste("[", Stations,
        "]", collapse = " , ")) %>%
    select(-c(Stations, vals)) %>%
    unique()

## Binding the row summed dataframe to
## FY database
station_details = cbind(miss_data$FY, station_details)

## Assigning the column names
colnames(station_details) = c("FY", "Total Stations")

## Merging the all the dataframes
merge(station_details, p, by = "FY", all = TRUE) %>%
    kable()
```

	Total	
FY	Stations	Station
FY_2019_2020	1	[IGSC.Planetarium.Complex]
FY_2020_2021	4	[Govt..High.School.Shikarpur] , [Muradpur] , [Rajbansi.Nagar] , [Samanpura]
FY_2021_2022	5	[DRM.Office.Danapur] , [Govt..High.School.Shikarpur] , [Muradpur] , [Rajbansi.Nagar] , [Samanpura]
FY_2022_2023	6	[DRM.Office.Danapur] , [Govt..High.School.Shikarpur] , [IGSC.Planetarium.Complex] , [Muradpur] , [Rajbansi.Nagar] , [Samanpura]

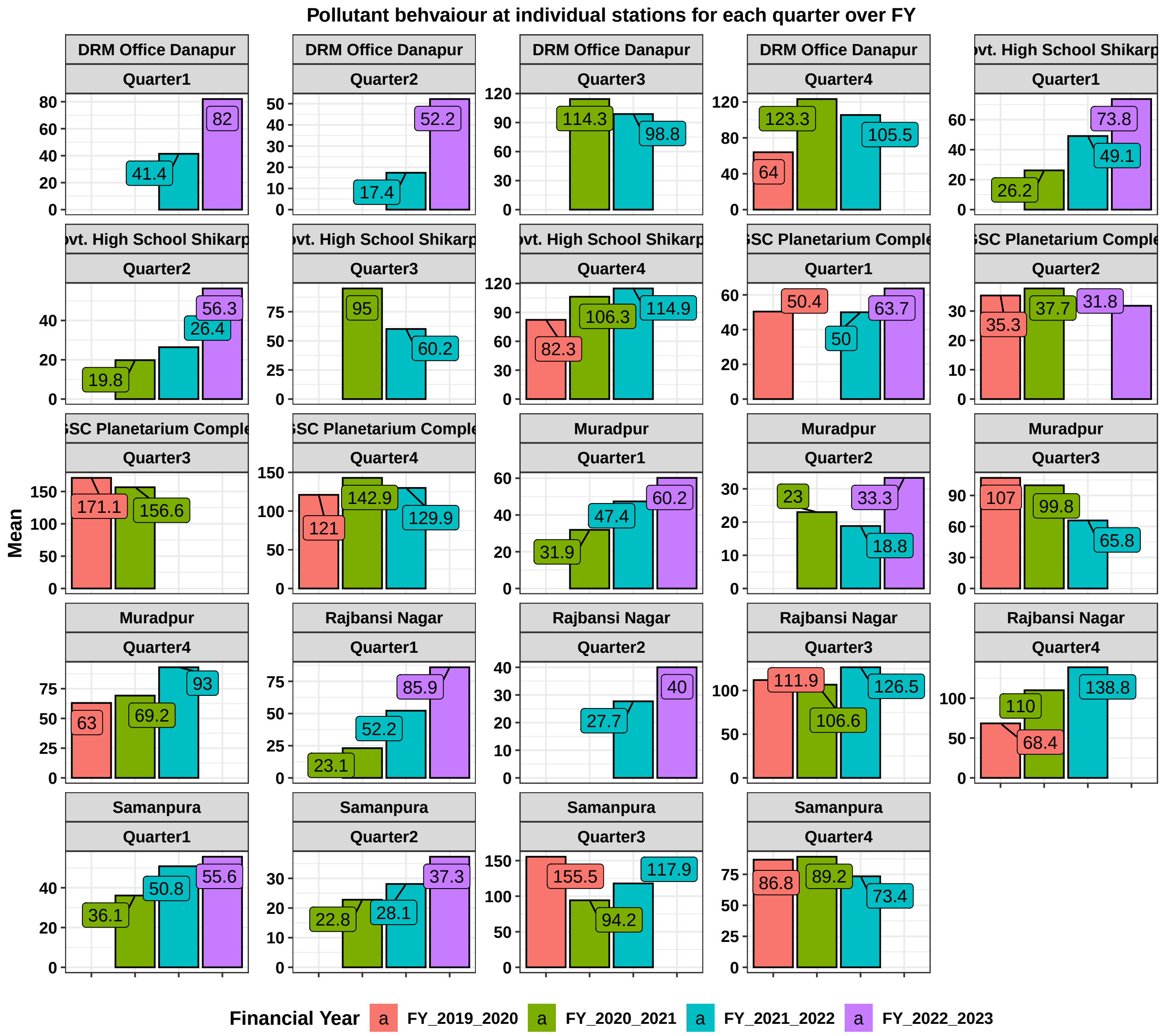
Let us find the overall PM trend over the subsequent FY for each stations. (Note Data unavailable for grea

```r
## Generating dataframe by performing
## mathematical calculation
Stationwise_FY = AQ_data_copy %>%
    left_join(AQ_data_FY %>%
        select(1, 3), by = "Date") %>%
    group_by(FY) %>%
    summarise_at(vars(-c(Date)), funs(mean(.,
        na.rm = TRUE))) %>%
    mutate_at(vars(-c(FY)), ~round(., digits = 2))

## Displaying the plot
Stationwise_FY %>%
    pivot_longer(cols = -c(FY), names_to = "Stations",
        values_to = "Mean") %>%
    left_join(miss_data %>%
        pivot_longer(cols = -c(FY), names_to = "Stations",
            values_to = "Missing_percent"),
        by = c("FY", "Stations")) %>%
    filter(Missing_percent <= 25) %>%
    ggplot(aes(x = FY, y = Mean)) + geom_point(shape = 21,
    size = 5, fill = "red", stroke = 1.5) +
    geom_line(color = "black", lty = 2, lwd = 1,
        aes(group = 1)) + facet_wrap(~Stations,
    ncol = 1, scales = "free_y") + geom_label_repel(aes(label = Mean,
    fontface = "bold"), size = 3.5) + ggtitle(label = "Pollutant behaviour across each stations over th
    theme_bw() + theme(axis.text = element_text(face = "bold",
    size = 10, color = "black"), strip.text = element_text(face = "bold",
    size = 10, color = "black"), plot.title = element_text(hjust = 0.5,
    face = "bold", size = 12, color = "black"))
```



Analyzing how the data retrieved by individual stations behaves each quarter over the FY.

```
AQ_data_copy %>%
    left_join(AQ_data_FY %>%
        select(1, 3), by = "Date") %>%
    FY_Quarters() %>%
    separate(col = Quarter_year, sep = "_",
        into = c("Quarter", "todel")) %>%
    select(-todel) %>%
    mutate_if(is.character, as.factor) %>%
    group_by(FY, Quarter) %>%
    summarise_at(vars(-c(Date)), funs(mean(.,
        na.rm = TRUE))) %>%
    mutate_at(vars(-c(FY, Quarter)), ~round(.,
        digits = 1)) %>%
    pivot_longer(!c(FY, Quarter, Month),
        names_to = "Stations", values_to = "Mean") %>%
    mutate_if(is.character, as.factor) %>%
    ggplot(aes(x = FY, y = Mean, fill = FY)) +
    geom_col(col = "black") + facet_wrap(Stations ~
    Quarter, scales = "free_y") + theme_bw() +
    scale_fill_discrete(name = "Financial Year") +
    ggtitle("Pollutant behvaiour at individual stations for each quarter over FY") +
    theme(axis.text.y = element_text(face = "bold",
        size = 10, color = "black"), axis.text.x = element_blank(),
        strip.text = element_text(face = "bold",
            size = 10, color = "black"),
        legend.text = element_text(face = "bold",
            size = 10, color = "black"),
        legend.title = element_text(face = "bold",
            size = 12, color = "black"),
        axis.title.y = element_text(face = "bold",
            size = 12, color = "black"),
        axis.title.x = element_blank(), legend.position = "bottom",
        plot.title = element_text(hjust = 0.5,
            face = "bold", size = 12, color = "black")) +
    geom_label_repel(aes(label = Mean))
```



Hope this R script was instrumental in analyzing the pollutant behaviour over the period across stations.