pt2

Xuan Nguyen

2024-11-12

install.packages(“readxl”)

```r
# Load necessary libraries
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
```

```r
library(tidyr)
library(readxl)
```

```r
# Load the data
data <- read_excel("/Users/xuanmn/Desktop/CSS 451/final project/pt2/who_aap_2021_v9_11august2022.xlsx",

# Filter data to keep only relevant columns and remove rows with missing years
data_filtered <- data %>%
  select(`WHO Region`, `City or Locality`, `Measurement Year`, `PM2.5 ( g/m3)`, `PM10 ( g/m3)`, `NO2 ( g/
         `PM25 temporal coverage (%)`, `PM10 temporal coverage (%)`, `NO2 temporal coverage (%)`) %>%
  filter(!is.na(`Measurement Year`))

# Calculate annual averages for each pollutant across all data
annual_avg <- data_filtered %>%
  group_by(`Measurement Year`) %>%
  summarise(across(starts_with("PM"), mean, na.rm = TRUE),
            `NO2 ( g/m3)` = mean(`NO2 ( g/m3)`, na.rm = TRUE))
```

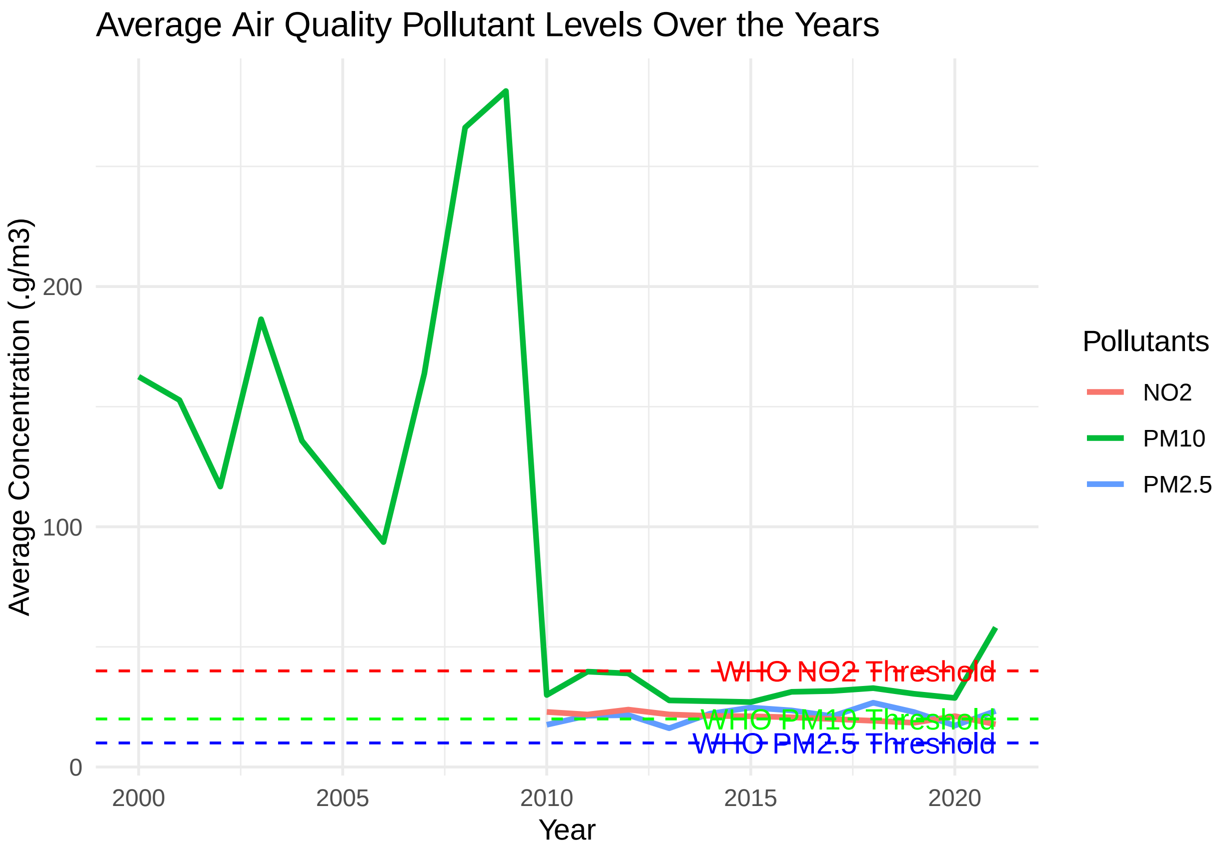
```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(starts_with("PM"), mean, na.rm = TRUE)`.
## i In group 1: `Measurement Year = 2000`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
```

```
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
# Plot the overall average air quality levels over the years
ggplot(annual_avg, aes(x = `Measurement Year`)) +
  geom_line(aes(y = `PM2.5 ( g/m3)`, color = "PM2.5"), size = 1) +
  geom_line(aes(y = `PM10 ( g/m3)`, color = "PM10"), size = 1) +
  geom_line(aes(y = `NO2 ( g/m3)`, color = "NO2"), size = 1) +
  labs(title = "Average Air Quality Pollutant Levels Over the Years",
       x = "Year",
       y = "Average Concentration ( g/m3)",
       color = "Pollutants") +
  geom_hline(yintercept = 10, linetype = "dashed", color = "blue", size = 0.5, show.legend = FALSE) +
  geom_hline(yintercept = 20, linetype = "dashed", color = "green", size = 0.5, show.legend = FALSE) +
  geom_hline(yintercept = 40, linetype = "dashed", color = "red", size = 0.5, show.legend = FALSE) +
  annotate("text", x = max(annual_avg$`Measurement Year`), y = 10, label = "WHO PM2.5 Threshold", color
  annotate("text", x = max(annual_avg$`Measurement Year`), y = 20, label = "WHO PM10 Threshold", color =
  annotate("text", x = max(annual_avg$`Measurement Year`), y = 40, label = "WHO NO2 Threshold", color =
  theme_minimal()
```

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

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 9 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

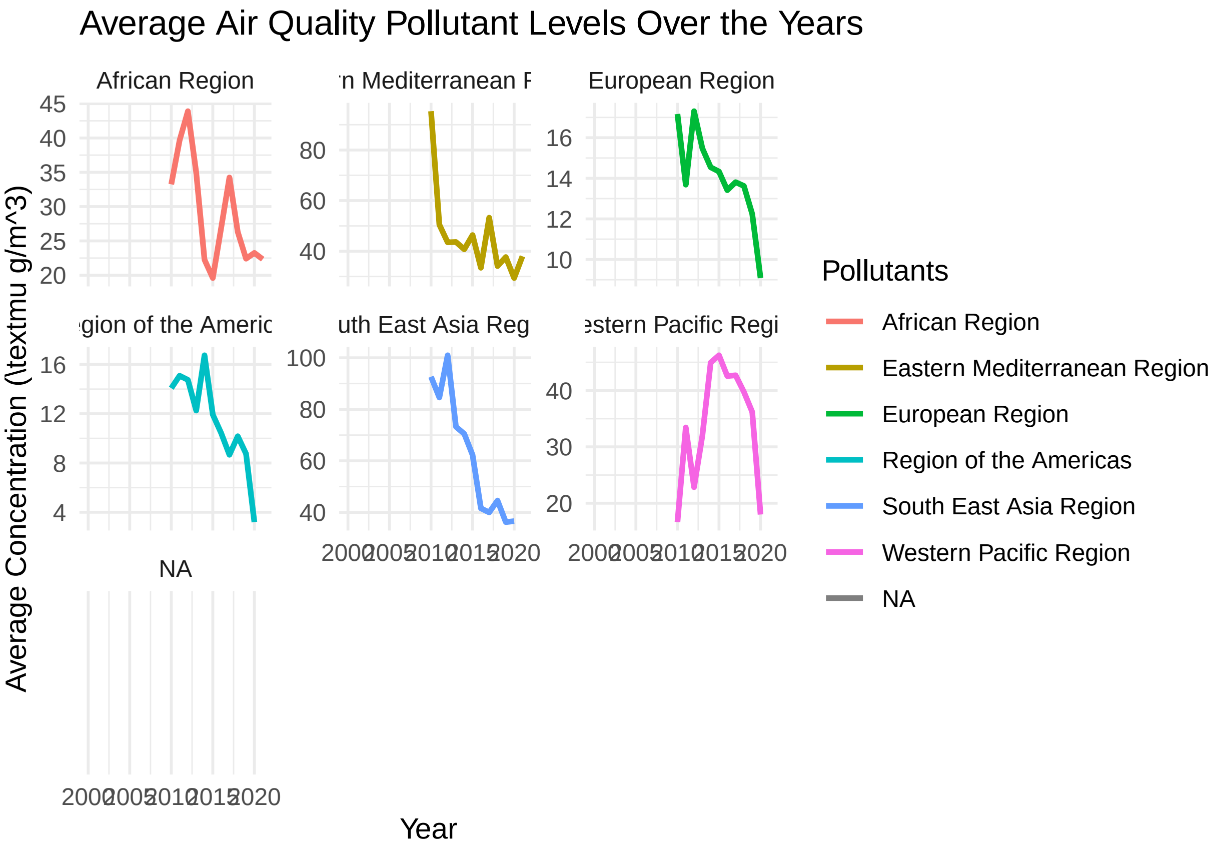


```r
# Regional comparison of pollutant levels over the years
regional_avg <- data_filtered %>%
  group_by(`WHO Region`, `Measurement Year`) %>%
  summarise(across(starts_with("PM"), mean, na.rm = TRUE),
            `NO2 ( g/m3)` = mean(`NO2 ( g/m3)`, na.rm = TRUE))
```

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

```r
ggplot(regional_avg, aes(x = `Measurement Year`)) +
  geom_line(aes(y = `PM2.5 ( g/m3)`, color = `WHO Region`), size = 1) +
  facet_wrap(~ `WHO Region`, scales = "free_y") +
labs(
  title = "Average Air Quality Pollutant Levels Over the Years",
  x = "Year",
  y = "Average Concentration (\\textmu g/m^3)", # Use \\textmu for
  color = "Pollutants"
)+
  theme_minimal()
```

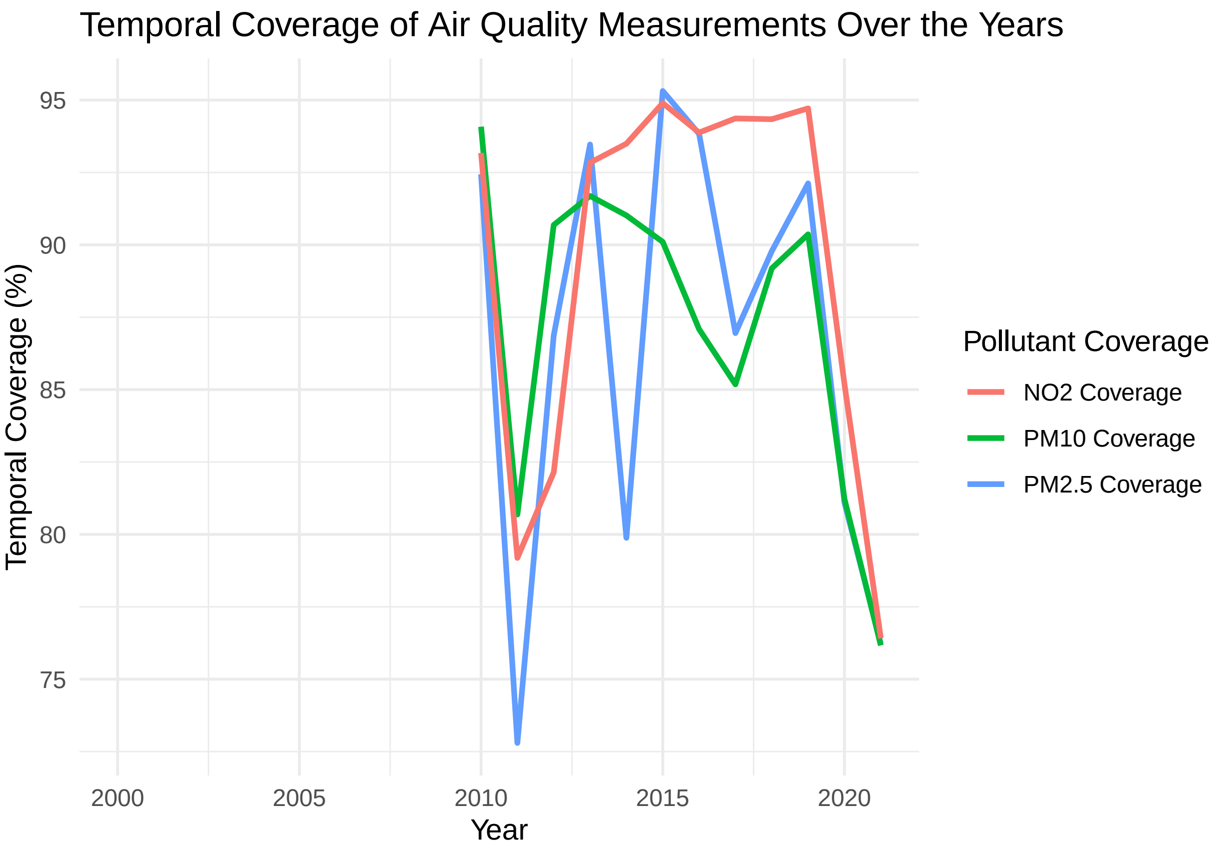
```
## Warning: Removed 10 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



```
# Temporal coverage visualization to assess data reliability over years
temporal_coverage <- data_filtered %>%
  group_by(`Measurement Year`) %>%
  summarise(`PM2.5 Coverage` = mean(`PM25 temporal coverage (%)`, na.rm = TRUE),
            `PM10 Coverage` = mean(`PM10 temporal coverage (%)`, na.rm = TRUE),
            `NO2 Coverage` = mean(`NO2 temporal coverage (%)`, na.rm = TRUE))

ggplot(temporal_coverage, aes(x = `Measurement Year`)) +
  geom_line(aes(y = `PM2.5 Coverage`, color = "PM2.5 Coverage"), size = 1) +
  geom_line(aes(y = `PM10 Coverage`, color = "PM10 Coverage"), size = 1) +
  geom_line(aes(y = `NO2 Coverage`, color = "NO2 Coverage"), size = 1) +
  labs(title = "Temporal Coverage of Air Quality Measurements Over the Years",
       x = "Year",
       y = "Temporal Coverage (%)",
       color = "Pollutant Coverage") +
  theme_minimal()
```

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 9 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 9 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



```r
# Correlation analysis between pollutants
cor_data <- data_filtered %>%
  select(`PM2.5 ( g/m3)`, `PM10 ( g/m3)`, `NO2 ( g/m3)`) %>%
  drop_na()

cor_matrix <- cor(cor_data, use = "complete.obs")
print(cor_matrix)
```

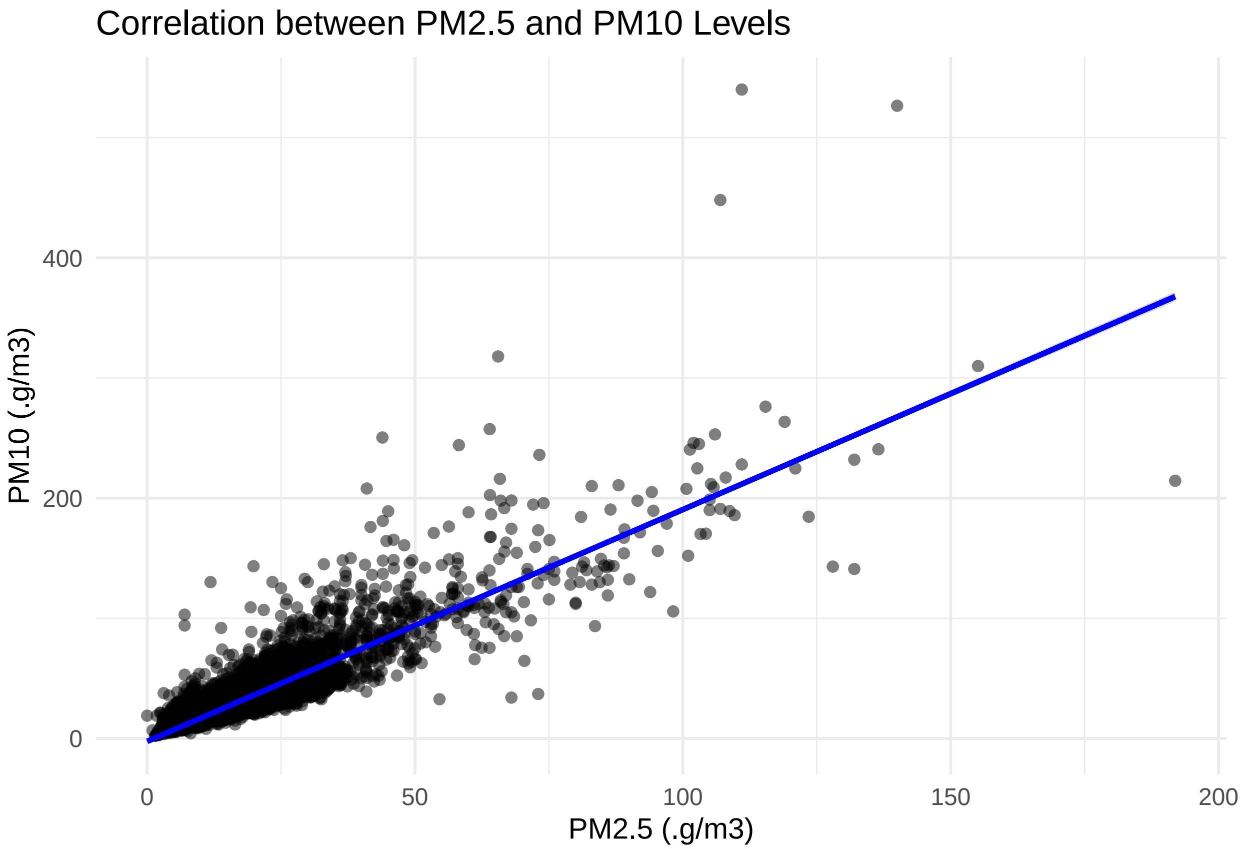
```
##                PM2.5 ( g/m3) PM10 ( g/m3) NO2 ( g/m3)
## PM2.5 ( g/m3)      1.0000000    0.8916884   0.3294467
## PM10 ( g/m3)       0.8916884    1.0000000   0.2801503
## NO2 ( g/m3)        0.3294467    0.2801503   1.0000000
```

```r
# Scatter plots for correlation between pollutants
ggplot(data_filtered, aes(x = `PM2.5 ( g/m3)`, y = `PM10 ( g/m3)`)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Correlation between PM2.5 and PM10 Levels",
       x = "PM2.5 ( g/m3)",
       y = "PM10 ( g/m3)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 23367 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 23367 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

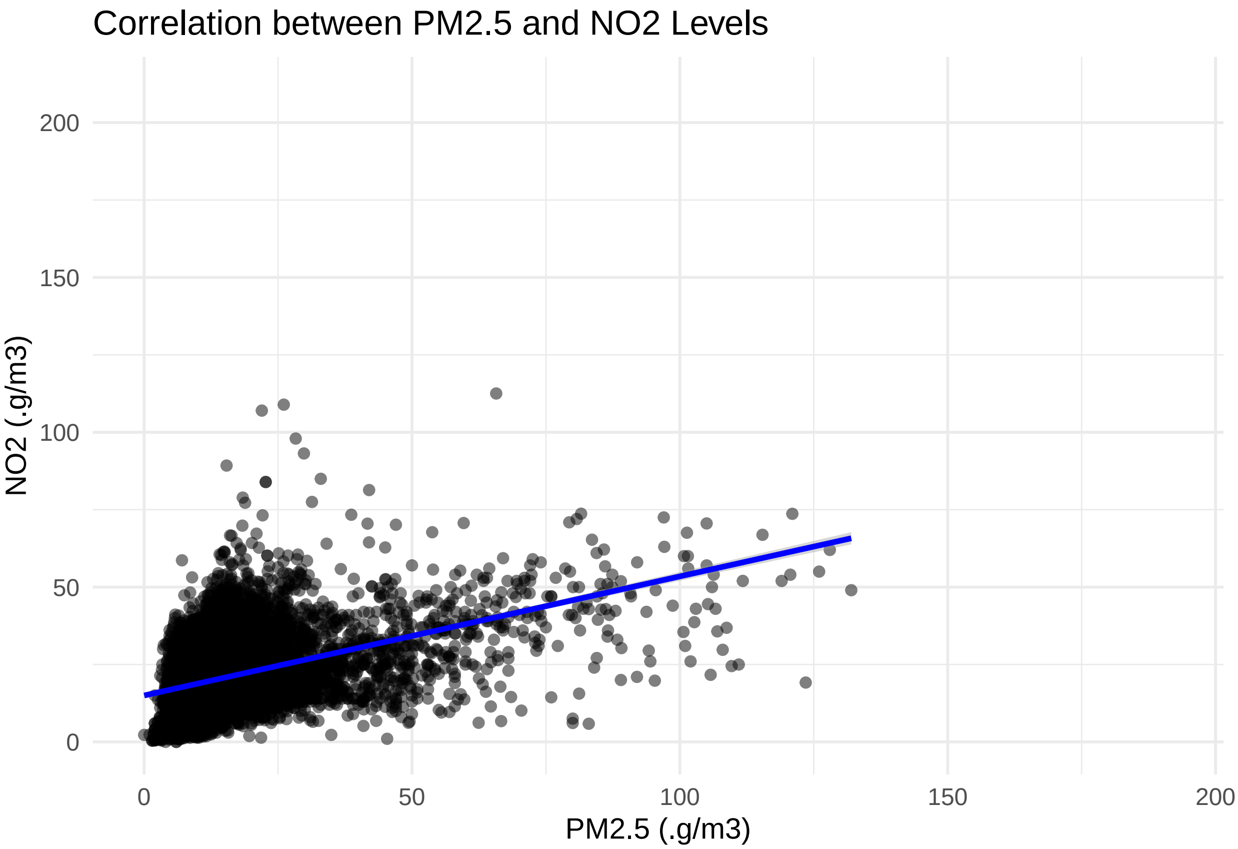


```r
ggplot(data_filtered, aes(x = `PM2.5 ( g/m3)`, y = `NO2 ( g/m3)`)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Correlation between PM2.5 and NO2 Levels",
       x = "PM2.5 ( g/m3)",
       y = "NO2 ( g/m3)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 23288 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 23288 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


Correlation between PM2.5 and NO2 Levels
NO2 (.g/m3)
200
150
100
50
0
PM2.5 (.g/m3)
0
50
100
150
200