Air Quality Analysis Using R: Visualization and Trend Analysis

Fatim Camara

12/16/2023

Link to Video Tutorial: https://youtu.be/FFfeXdOp4Lk

Link to R script: https://gist.github.com/fermela2024/517fd62bb55a79403358ebd0449e451d

Source of the Dataset: https://github.com/fermela2024/AQI-CSV-files-

Website where the data set was obtained: https://www.epa.gov/outdoor-air-quality-data/download-daily-data.

Purpose:

This manual offers a comprehensive method for analyzing air quality trends in Hawaii, New York, and Massachusetts throughout 2023 using R. This tailored script will delve into the specifics of PM2.5 levels, leveraging R's statistical tools to generate plots that reveal monthly variations and comparisons. The objective is to utilize R's capabilities to understand the air quality landscape over a single year, which can be pivotal for environmental analysis and informing regional air quality improvements.

Background:

The Air Quality Index (AQI) is an essential indicator of air pollution and related health risks. Its importance escalates amidst rapid urbanization and industrial advancement, where it becomes vital for the protection of public health. Countries employ their unique standards for AQI to alert their populations about immediate and serious health dangers due to air pollution.

In particular, monitoring particulate matter 2.5 (PM2.5) levels is crucial for gauging the state of air pollution and understanding its health ramifications in urban environments. As we grapple with environmental challenges, such scrutiny becomes ever more pertinent.

Air quality exerts a significant influence on health and the ecological landscape. Simulating AQI trends enables us to anticipate shifts in urban air quality and assess the effectiveness of environmental policies. Utilizing R for these analyses transcends basic programming; it underscores R's proficiency in navigating and interpreting complex environmental phenomena. Although the data used in this R script are simulated, they reflect the variability and trends one might observe in real-world conditions, providing a foundation for developing strategies to promote environmental health.

Tool/Dataset Description:

For our analysis, I have sourced authentic data directly from the Environmental Protection Agency (EPA) using their "Download Daily Data" tool available at https://www.epa.gov/outdoor-air-quality-data/download-daily-data. By applying filters based on the pollutant of interest and geographic location, we can obtain comprehensive datasets for all counties within the selected regions. The resultant CSV files provide us with real-world air quality measurements, specifically PM2.5 concentrations, for the year 2023. This data forms the backbone of our analysis, allowing us to apply R's data manipulation capabilities via packages like "dplyr" and leverage "ggplot2" for a detailed graphical representation of the air quality trends across different cities.

Table of Contents

R Script Guide: Step-by-Step Process

This guide will instruct on using R to process, analyze, and visualize PM2.5 data. For detailed steps, consult the provided script sections. We start by installing and loading necessary packages, then proceed to generate individual scatter plots and comparative box plots. We conduct ANOVA for statistical comparison and culminate with trend analysis across all cities, visualizing the data in a combined plot to discern broader patterns.

This script serves as a thorough guide for analyzing PM2.5 air pollution data using R, detailing each step from setting up the necessary environment to visualizing and interpreting the data.

Part 1: Preparing the R Environment

a) Setting Up Required R Packages to conduct Analysis

```
#Part 1
#Setting Up the Environment: Installing and Loading Essential R Packages"
# a) Installing Required Packages
install.packages("ggplot2")
install.packages("dplyr")
install.packages("lubridate")
install.packages("readr")
install.packages("scales")
```

To commence data analysis in RStudio, it's crucial to first install a set of key R packages from the Comprehensive R Archive Network (CRAN). These packages are not pre-installed in RStudio, which is an integrated development environment (IDE) for R. However, RStudio simplifies the process of installing and managing these essential packages. The key packages include:

- **'ggplot2':** A powerful tool for creating advanced and visually appealing graphics.
- **'dplyr':** Essential for streamlined data manipulation, making data handling tasks more efficient.
- **'lubridate':** This package simplifies working with date and time data in R.
- **'readr':** Enhances the efficiency of reading and writing data, an integral part of data handling.
- **'scales':** Useful for customizing plot axes and scales, allowing for detailed adjustments in data visualization.

For detailed information on each package, refer to the following links:

- **ggplot2**: CRAN - Package ggplot2
- **dplyr**: CRAN - Package dplyr
- **lubridate**: CRAN - Package lubridate
- **readr**: CRAN - Package readr
- **scales**: CRAN - Package scales

b) Loading Libraries into RStudio

```
#Part 1
#Setting Up the Environment: Installing and Loading Essential R Packages"
# a) Installing Required Packages
install.packages("ggplot2")
install.packages("dplyr")
install.packages("lubridate")
install.packages("readr")
install.packages("scales")
```

After installing these packages, the next step is to load them into your current R session. This is done using the **library() function.** Loading these libraries is necessary to utilize their wide range of functionalities for data analysis and visualization within RStudio.

For detailed information about library() function refer to the following link:

[R library() function documentation]

Part 2: Creating Individual State Scatter Plots

This section of the script focuses on creating individual scatter plots that depict PM2.5 concentrations for various cities, segmented by year and month.

```
# Part 2 Generate Individual scatter plots PM2.5 by city, year, and months
# a) Function to generate and visualize the plot for a single state
generate_plot <- function(data, state_name, thresholds) {
  data$Date <- as.Date(data$Date, format = "%m/%d/%Y")
  data$Year <- year(data$Date)
  data$State <- as.factor(data$STATE)

  # b) To add a column to the data to indicate if the value is above the threshold
  data$Threshold <- ifelse(data$`Daily Mean PM2.5 Concentration` > thresholds[2], 'Above', 'Below')

  plot_title <- paste("Daily Mean PM2.5 Concentration for", state_name, "in 2023")

  # c) To generate monthly breaks based on the year of the data
  year_data <- unique(data$Year)[1]
  monthly_breaks <- seq(as.Date(paste(year_data, "-01-01", sep="")),
                        as.Date(paste(year_data, "-12-01", sep="")), by="1 month")

  # d) To create a plot with aes for color and linetype for the legend
  p <- ggplot(data, aes(x = Date, y = `Daily Mean PM2.5 Concentration`)) +
    geom_point(aes(color = Threshold), alpha = 0.6) +
    geom_smooth(method = "lm", se = FALSE, color = "black") +
    geom_hline(aes(yintercept = thresholds[1], linetype = "Annual Limit", color = "Annual Limit"), size = 1) +
    geom_hline(aes(yintercept = thresholds[2], linetype = "24-hour Limit", color = "24-hour Limit"), size = 1) +
    labs(
      title = plot_title,
      x = "Month",
      y = "Daily Mean PM2.5 Concentration (µg/m³)",
      color = "Key",
      linetype = "Key"
    ) +
    scale_x_date(labels = date_format("%b"), breaks = monthly_breaks) +
    scale_color_manual(values = c('Below' = 'blue', 'Above' = 'orange', 'Annual Limit' = 'red', '24-hour Limit' = 'green')) +
    scale_linetype_manual(values = c('Annual Limit' = 'dashed', '24-hour Limit' = 'dotted')) +
    theme_minimal() +
    theme(
      legend.position = "bottom",
      axis.text.x = element_text(angle = 90, hjust = 1),
      legend.title.align = 0.5,
      legend.box = "horizontal"
    )
```

```r
  # e) To print/Display the plot in RStudio
  print(p)

  # f) Save the plot as an image file
  ggsave(paste0(state_name, "_PM2.5_2023_plot.png"), plot = p, width = 12, height = 6, dpi = 300)
}

# g) To set thresholds based on the WHO Air Quality Guidelines for tresholds for PM2.5
thresholds <- c(5, 15) # Annual and 24-hour thresholds

# h) To load data for each state individually on your computer you can update the directory for the files
# Scatter plot for Hawaii
hawaii_data <- read_csv("/Users/fatimcamara/Hawaii2023.csv", show_col_types = FALSE)
generate_plot(hawaii_data, "Hawaii", thresholds)

# Scatter plot for Massachusetts
mass_data <- read_csv("/Users/fatimcamara/Massachusetts2023.csv", show_col_types = FALSE)
generate_plot(mass_data, "Massachusetts", thresholds)

# Scatter plot for New york
newyork_data <- read_csv("/Users/fatimcamara/NewYork2023.csv", show_col_types = FALSE)
generate_plot(newyork_data, "New York", thresholds)
```

a) **Function Definition (generate_plot):**
   - This function, **generate_plot**, is defined to create scatter plots. It takes three parameters: **data** (the dataset), **state_name** (name of the state for which the plot is generated), and **thresholds** (PM2.5 concentration limits).

     For additional information about the functions used in this script, please refer to the following hyperlinks:

   - R Functions

b) **Data Preprocessing within the Function:**
   - **as.Date(data$Date, format = "%m/%d/%Y")**: Converts the date column to R's Date type.
   - **year(data$Date)**: Extracts the year from the Date using the **lubridate** package.
   - **as.factor(data$STATE)**: Converts the state column to a factor (categorical variable).

     For additional information about the functions used in this script, please refer to the following hyperlinks:
   - as.Date
   - lubridate's year
   - as.factor

c) **Threshold Analysis and Plot Customization:**
   - **ifelse(data$'Daily Mean PM2.5 Concentration' > thresholds[2], 'Above', 'Below')**: Creates a new column to categorize data points as 'Above' or 'Below' the threshold.
   - **ggplot** and associated functions (**geom_point**, **geom_smooth**, **geom_hline**, etc.) are used to create the scatter plot with a trend line and threshold lines.
   - **scale_x_date**, **scale_color_manual**, **scale_linetype_manual**: Customize the x-axis, color, and line types in the plot.

     For additional information about the functions used in this script, please refer to the following hyperlinks:
   - ggplot2

- ifelse

d) **Printing and Saving the Plot:**
   - **print(p)**: Displays the generated plot in RStudio.
   - **ggsave(...)**: Saves the plot as an image file.

   For additional information about the functions used in this script, please refer to the following hyperlinks:

   - ggsave

e) **Setting Thresholds and Loading Data:**
   - **thresholds <- c(5, 15)**: Sets the PM2.5 concentration thresholds based on guidelines.
   - **read_csv(...)**: Reads the CSV files for each city (Hawaii, Massachusetts, New York) and generates plots for each.

   For additional information about the functions used in this script, please refer to the following hyperlinks:
   - readr's read_csv

f) **ggsave() Function**:

   - This function is used to save a plot created using ggplot2. It allows specifying the file name, plot object, dimensions, and resolution of the saved plot.

   For additional information about the functions used in this script, please refer to the following hyperlinks:

   - Hyperlink: ggsave function documentation

g) **Setting Thresholds**:

   - These are predefined limits set to categorize the PM2.5 data. In this script, two thresholds (5 and 15) are defined, likely representing safe and unsafe levels of PM2.5 based on WHO guidelines.

   For additional information about the threshold used in this script, please refer to the following hyperlinks:

   - WHO Air Quality Guidelines

h) **read_csv() Function**:

   - This function is used to read a CSV file into R as a data frame. It's part of the readr package and is known for its speed and simplicity.

   For additional information about the functions used in this script, please refer to the following hyperlinks:

- https://www.geeksforgeeks.org/read-contents-of-a-csv-file-in-r-programming-read-csv-function/

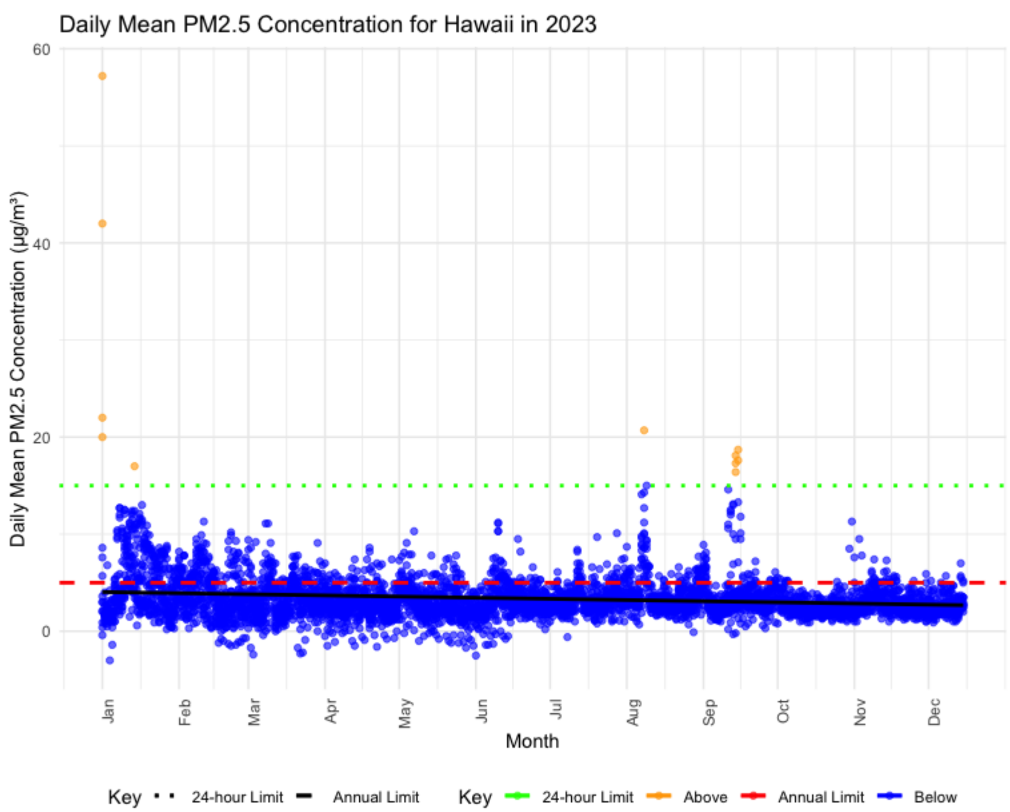
i) **generate_plot() Function**:

- The function defined in the script is to create scatter plots. It processes the data, applies conditional formatting based on thresholds, and generates a plot using ggplot2.
- *Hyperlink* for Custom Functions in R: Writing Functions in R
- *Hyperlink* for ggplot2: ggplot2 package documentation

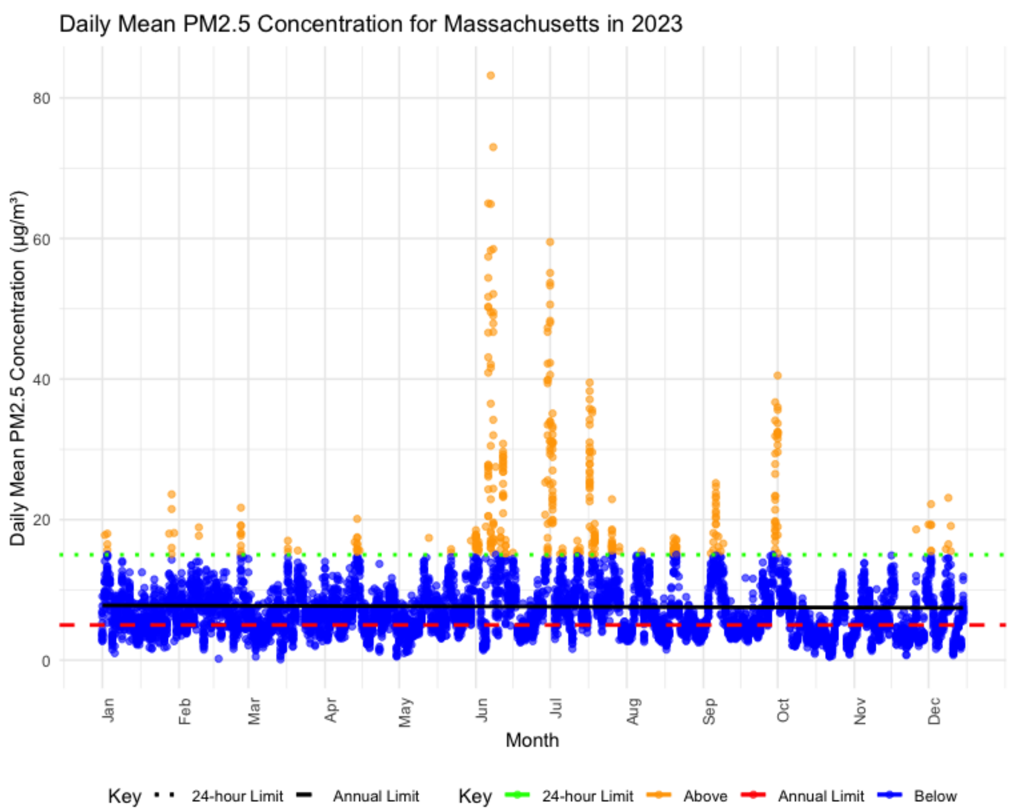
j) **ggplot() Function**:

- The primary function of the ggplot2 package, is used to initialize a ggplot object. It sets up the data and aesthetics (aes) of the plot.
- Hyperlink: ggplot function documentation

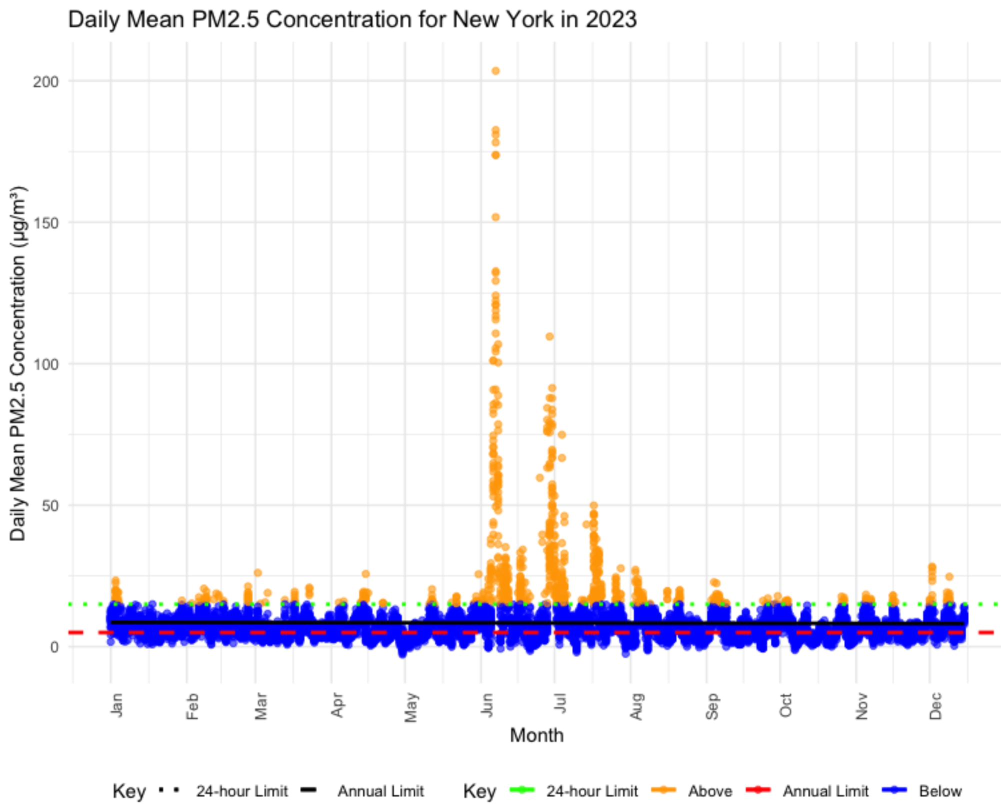
Plots generated by script:



The scatter plot represents Hawaii's daily PM2.5 levels for 2023, with most values being low. Two thresholds are marked: a red dashed line for the annual limit and a green dotted line for the 24-hour limit. Data points are colored blue for concentrations below and orange for those exceeding the 24-hour limit, with occasional peaks suggesting days of higher pollution. The months from January to December are on the x-axis, while the y-axis measures PM2.5 levels. The legend at the bottom decodes colors and line types.



This scatter plot illustrates the daily PM2.5 levels in Massachusetts for the year 2023. The plot shows a baseline of values with sporadic spikes, indicating days with higher PM2.5 levels. Two reference lines are present: the annual limit as a red dashed line and the 24-hour limit as a green dotted line. The data points are predominantly below the 24-hour limit, depicted in blue, while the points exceeding this threshold are in orange, suggesting occasional pollution surges. The x-axis represents the months from January through December, and the y-axis quantifies the PM2.5 concentration.



In the plot for New York's daily mean PM2.5 concentration in 2023, there is a noticeable trend of values consistently staying below the 24-hour limit threshold for most of the year, with a large number of data points concentrated close to zero, indicating many days with low PM2.5 levels. However, there is a prominent spike in PM2.5 concentration around mid-year, where the values exceed the 24-hour limit significantly. This suggests a period of increased air pollution. The rest of the year appears to show relatively stable and low PM2.5 concentrations, with occasional smaller peaks that do not reach the annual limit.

Part 3: Comparative Analysis with Box Plots

The script for Part 3 is about creating a comparative box plot to analyze the PM2.5 levels across different cities and conducting statistical tests to understand if there are significant differences.

```r
#Part 3 Comparative Box Plot Analysis
# a) To load data for each state individually and add a 'City' column
#Data set for Hawaii
hawaii_data <- read_csv("/Users/fatimcamara/Hawaii2023.csv", show_col_types = FALSE) %>%
  mutate(City = 'Hawaii')

#Data set for Massachusetts
mass_data <- read_csv("/Users/fatimcamara/Massachusetts2023.csv", show_col_types = FALSE) %>%
  mutate(City = 'Massachusetts')

#Data set for New york
newyork_data <- read_csv("/Users/fatimcamara/NewYork2023.csv", show_col_types = FALSE) %>%
  mutate(City = 'New York')

# b) To combine the data into one data frame
all_data <- bind_rows(hawaii_data, mass_data, newyork_data) %>%
  mutate(Date = as.Date(Date, format = "%m/%d/%Y"),
         Year = as.factor(year(Date)), # Convert Year to a factor for the ANOVA
         City = as.factor(City)) # Ensure City is a factor

# d) To create box plot comparing Daily Mean PM2.5 Concentration for each city
p <- ggplot(all_data, aes(x = City, y = `Daily Mean PM2.5 Concentration`, fill = City)) +
  geom_boxplot() +
  facet_wrap(~Year, scales = 'free_x') + # Facet by Year
  labs(title = "Comparison of Daily Mean PM2.5 Concentration by City",
       x = "City",
       y = "Daily Mean PM2.5 Concentration (µg/m³)") +
  theme_minimal() +
  theme(legend.position = "none") # Hide the legend for city colors

# e) To print/Display the box plot in RStudio
print(p)

# f) To save the box plot as an image file
ggsave("PM2.5_comparison_plot.png", plot = p, width = 12, height = 6, dpi = 300)

# Part 4 Statistical Analysis: ANOVA and Post-Hoc Testing for City Comparisons

# a) Statistical test (ANOVA) to assess differences across cities
anova_results <- aov(`Daily Mean PM2.5 Concentration` ~ City, data = all_data)
summary(anova_results)

# If the ANOVA is significant, proceed with a post-hoc test to find where the differences lie
if (summary(anova_results)[[1]]$'Pr(>F)'[1] < 0.05) {
  post_hoc <- TukeyHSD(anova_results)
  print(post_hoc)
}
```

a) **Loading Data**: Individual datasets for Hawaii, Massachusetts, and New York are loaded using the **read_csv()** function from the **readr** package. For each dataset, a new column named 'City' is added to label the data accordingly.

   - Hyperlink : [read_csv() documentation]

b) **Combining Data**: The datasets for each city are combined into a single data frame using the **bind_rows()** function from the **dplyr** package. This merged data is then modified to ensure the 'Date' column is in the proper date format and that 'Year' and 'City' are recognized as categorical factors, which is necessary for statistical analysis.

   - Hyperlink : [bind_rows() documentation]

- Hyperlink : [mutate() documentation]

c) **Creating Box Plots**: A box plot for the combined data is created using **ggplot()** from the **ggplot2** package, comparing PM2.5 levels by city. The **geom_boxplot()** function is used to draw the box plot, and **facet_wrap()** is applied to create separate plots for each year if the data spans multiple years.

- Hyperlink : [ggplot() documentation]
- Hyperlink : [geom_boxplot() documentation]

d) **Saving Plots**: The **ggsave()** function saves the created box plot as an image file on the computer.

- Hyperlink : [ggsave() documentation]

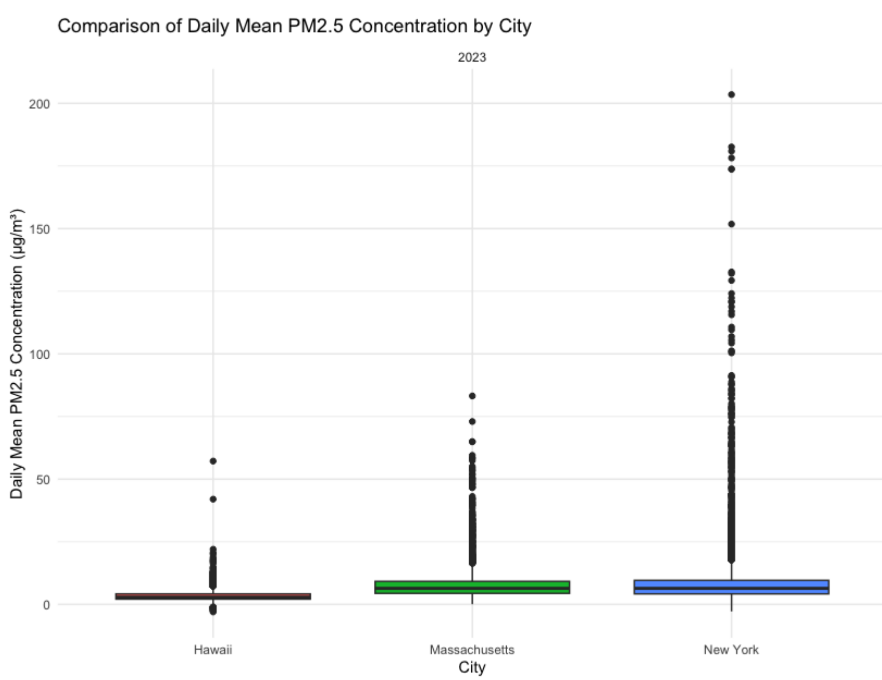
e) **Statistical Analysis**: An Analysis of Variance (ANOVA) is performed to test if there are statistically significant differences in PM2.5 levels between the cities. This is done using the **aov()** function.

- Hyperlink : [aov() documentation]

f) **Post-Hoc Testing**: If the ANOVA results indicate significant differences, a post-hoc test (Tukey's Honest Significant Difference test) is conducted using **TukeyHSD()**. This test identifies which specific groups (cities, in this case) differ from each other.

- Hyperlink : [TukeyHSD() documentation]

Plots and statistical data generated by script, see below for output:



The box plot presents a comparative analysis of daily mean PM2.5 concentrations among Hawaii, Massachusetts, and New York for the year 2023. Each box delineates the middle 50% of values, known as the interquartile range (IQR), with the internal line representing the median PM2.5 level. Outliers, depicted as points beyond the boxes, represent measurements significantly divergent from the norm. Hawaii's data cluster tightly around a low median, indicating consistently cleaner air. Massachusetts has a broader IQR, suggesting greater variation in air quality, while New York's plot reveals the most pronounced variability, with numerous outliers pointing to episodic spikes in PM2.5 concentration, hinting at potentially poorer air quality. Overall, the plot suggests Hawaii enjoys the most stable and clean air, while New York faces challenges with air pollution.

Summary of "anova_results"

```
> # a) Statistical test (ANOVA) to assess differences across cities
> anova_results <- aov(`Daily Mean PM2.5 Concentration` ~ City, data = all_data)
> summary(anova_results)
               Df  Sum Sq Mean Sq F value Pr(>F)
City            2   86828   43414   758.1 <2e-16 ***
Residuals   21769 1246657      57
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> # If the ANOVA is significant, proceed with a post-hoc test to find where the differences lie
> if (summary(anova_results)[[1]]$'Pr(>F)'[1] < 0.05) {
+   post_hoc <- TukeyHSD(anova_results)
+   print(post_hoc)
+ }
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = `Daily Mean PM2.5 Concentration` ~ City, data = all_data)

$City
                             diff       lwr      upr p adj
Massachusetts-Hawaii    4.2274842 3.8972503 4.557718     0
New York-Hawaii         4.9835787 4.6766352 5.290522     0
New York-Massachusetts 0.7560945 0.4768968 1.035292     0
```

The **summary(anova_results)** provides a statistical summary of the Analysis of Variance (ANOVA) test performed on the daily mean PM2.5 concentration data across different cities. Here's a breakdown of the summary output:

- **Df** (Degrees of Freedom): Represents the number of levels in the factor variable minus one. For **City**, which has three levels (Hawaii, Massachusetts, and New York), the degrees of freedom are 2. For "**Residuals"**, which represents the variation not explained by the model, it is the total number of observations minus the number of levels in the factor.
- **Sum Sq** (Sum of Squares): This column shows the total variation for the **City** factor and the "**Residuals"**. For the city, it is the variation in PM2.5 levels that can be attributed to

differences between the cities, and for the residuals, it's the variation that is not explained by the city differences.

- **Mean Sq** (Mean Square): This is the sum of squares divided by the corresponding degrees of freedom. It represents the average variation for the **City** factor and **Residuals**.
- **F value**: This is the ratio of the Mean Square of the **City** to the Mean Square of the **Residuals**. A higher F value indicates a greater variance between the groups than within the groups, which often signals a significant effect of the factor being tested.
- **Pr(>F)** (p-value): This indicates the probability of observing the F value, at least as extreme as the one calculated, if there were no real differences between the cities' mean PM2.5 concentrations. A p-value less than 0.05 is commonly used as a threshold for statistical significance.

The asterisks (*****) next to the p-value represent the level of significance. In this case, with a p-value of **<2e-16**, it is extremely significant, meaning it is highly unlikely that the observed differences are due to chance.

Since the ANOVA indicated significant differences, a Tukey Honest Significant Difference (HSD) post-hoc test was conducted to compare the means between each pair of cities. The post-hoc results show:

- **Massachusetts-Hawaii**: A mean difference of approximately 4.23 µg/m³, with a p-value of 0, indicating that the PM2.5 levels are significantly different between these two cities.
- **New York-Hawaii**: A mean difference of approximately 4.98 µg/m³, with a p-value of 0, again indicating a significant difference between these cities.
- **New York-Massachusetts**: A smaller mean difference of approximately 0.76 µg/m³, but still with a p-value of 0, suggesting a significant difference in PM2.5 levels.

The **diff** is the estimated difference between the means of the groups, **lwr,** and **upr** are the lower and upper limits of the 95% confidence interval for the difference, and **p adj** is the adjusted p-value for multiple comparisons. The zero p-values indicate that the differences are significant beyond the usual thresholds for statistical significance.

For more detailed information on ANOVA and the Tukey HSD test, you can refer to the hyperlinks provided below:

- ANOVA in R
- Tukey HSD Test

Part 4: Trend Analysis and Visualization

```r
#Part 4 Trend Analysis for Each City

# a) To generate and visualize the trend plot for a single city
generate_trend_plot <- function(data, city_name) {
  # Prepare data: convert Date to Date type and extract Month and Year
  prepared_data <- data %>%
    mutate(Date = as.Date(Date, format = "%m/%d/%Y"),
           Month = factor(format(Date, "%b"), levels = month.abb),
           Year = year(Date)) %>%
    filter(Year == 2023) # Filter for the year 2023

  # b) To plot the trend line for the given city
  trend_plot <- ggplot(prepared_data, aes(x = Month, y = `Daily Mean PM2.5 Concentration`, group = 1)) +
    geom_line(stat = "summary", fun = mean, color = "blue", size = 1) +
    geom_point(color = "red", size = 2) +
    theme_minimal() +
    labs(
      title = paste("Trend of PM2.5 Levels in 2023 by Month for", city_name),
      x = "Month", y = "Daily Mean PM2.5 Concentration (µg/m³)"
    ) +
    theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
    # c) To add a manual legend at the bottom
    scale_color_manual(name = "",
                       values = c("mean" = "blue", "data" = "red"),
                       labels = c("Mean PM2.5 Concentration", "Daily PM2.5 Data Points")) +
    guides(color = guide_legend(title.position = "top", title.hjust = 0.5, ncol = 2, byrow = TRUE)) +
    theme(legend.position = "bottom")

  # d) To Print/Diaplay the plot in RStudio
  print(trend_plot)

  # e) To save the plot as an image file
  ggsave(paste0(city_name, "_PM2.5_2023_trend_plot.png"), plot = trend_plot, width = 12, height = 6, dpi = 300)
}

# Load data for each city individually
hawaii_data <- read_csv("/Users/fatimcamara/Hawaii2023.csv", show_col_types = FALSE)
mass_data <- read_csv("/Users/fatimcamara/Massachusetts2023.csv", show_col_types = FALSE)
newyork_data <- read_csv("/Users/fatimcamara/NewYork2023.csv", show_col_types = FALSE)

# Generate and visualize trend plots for each city
generate_trend_plot(hawaii_data, "Hawaii")
generate_trend_plot(mass_data, "Massachusetts")
generate_trend_plot(newyork_data, "New York")
```

The **generate_trend_plot** function in R creates a visual trend line of PM2.5 levels for a specified city's data in the year 2023. It involves the following steps:

a) **Data Preparation**: The **Date** column is converted to a Date object, and new columns for **Month** and **Year** are created using the **mutate** function. The **factor** function is used to order the months correctly, and **year** extracts the year from the date.

b) **Trend Plotting**: The **ggplot** function constructs the trend plot. It uses **geom_line** to draw the trend line, representing the mean PM2.5 concentration over each month, and **geom_point** to plot individual data points.

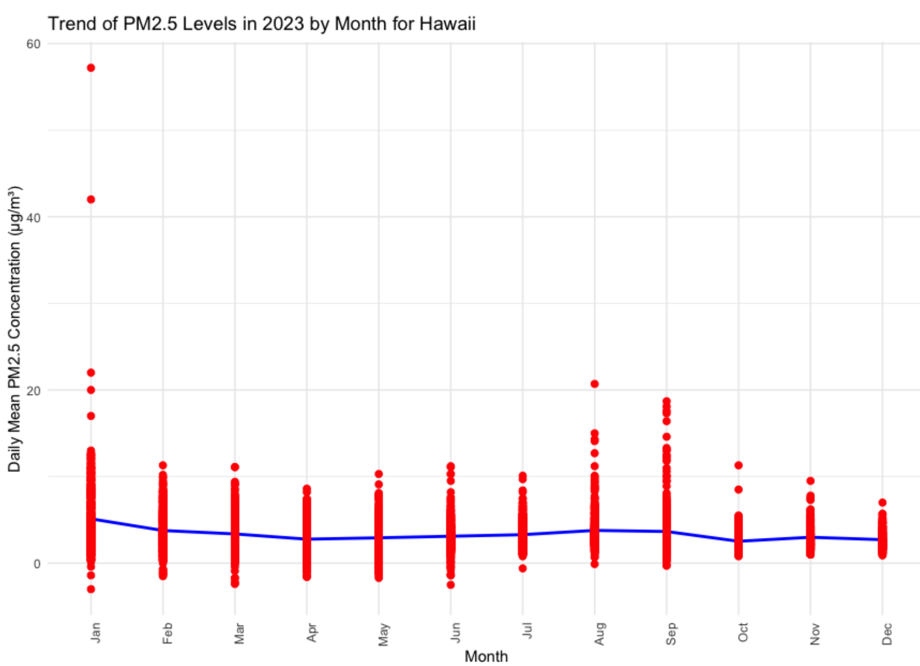
c) **Legend Addition**: A manual legend is added using **scale_color_manual**, which differentiates between the mean concentration trend (in blue) and the individual data points (in red).

d) **Printing and Saving**: The plots are displayed in RStudio using **print**, and saved to a file with **ggsave**.

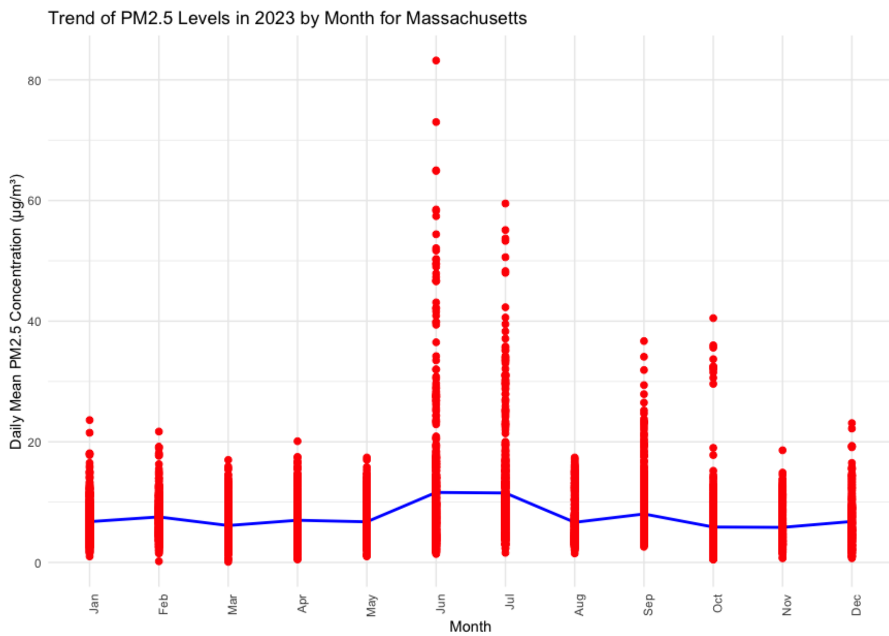
For additional information about the functions used in this script, please refer to the following hyperlinks:

- **mutate** from dplyr
- **ggplot** from ggplot2
- **geom_line** from ggplot2
- **geom_point** from ggplot2
- **scale_color_manual** from ggplot2
- **guides** from ggplot2
- **ggsave** from ggplot2
- **read_csv** from readr

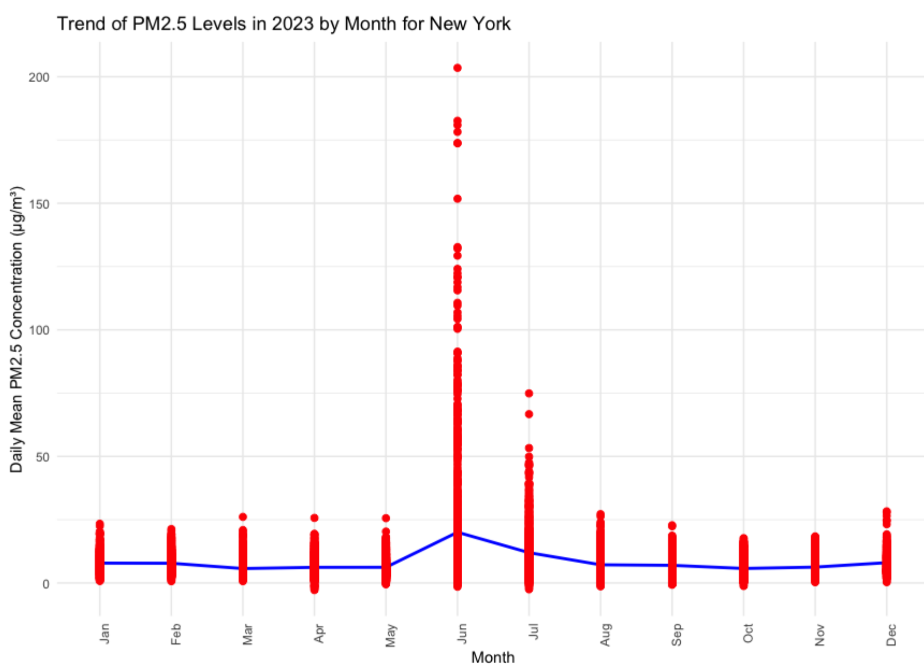
Plots generated by the script:



The plot displays the monthly trend of PM2.5 air pollutant levels for Hawaii in the year 2023. The red dots represent daily mean PM2.5 concentrations and the blue line indicates the average trend over the year. The concentration of PM2.5 varies throughout the months, with several noticeable spikes suggesting episodic increases in pollution levels. Overall, there is a clear fluctuation in daily values, but the mean trend line remains relatively stable across the months.



The trend plot for Massachusetts in 2023 shows daily PM2.5 levels with significant daily fluctuations, especially in the middle months where pronounced spikes are indicating higher pollution events. The average trend remains relatively steady throughout the year, despite these variations.



The trend plot for New York in 2023 indicates that while the daily mean PM2.5 concentration generally stays within a moderate range, notable peaks are suggesting occasional high pollution events, particularly in the middle of the year. These spikes are much higher compared to the overall trend, which could point to specific episodes that dramatically worsen air quality.

Part 5: Combined Trend Analysis

This section of the script is designed to create a visual trend analysis of PM2.5 levels across multiple cities for the year 2023.

```r


#Part 5 - Combined Trend Plot for All Cities

# To combine all the data
generate_combined_trend_plot <- function(all_data) {
  # a) Filter for the year 2023
  data_2023 <- all_data %>%
    filter(Year == 2023) %>%
    mutate(Month = factor(format(Date, "%b"), levels = month.abb))

  # b) Plotting the trend line for each city
  combined_trend_plot <- ggplot(data_2023, aes(x = Month, y = `Daily Mean PM2.5 Concentration`, group = City, color = City)) +
    geom_line(stat = "summary", fun = mean) +
    geom_point(aes(shape = City)) +
    scale_shape_manual(values = c(16, 17, 18)) +
    theme_minimal() +
    labs(
      title = "Trend of PM2.5 Levels in 2023 by Month Across Cities",
      x = "Month", y = "Daily Mean PM2.5 Concentration (µg/m³)",
      color = "City",
      shape = "City"
    ) +
    theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
    # c) To place the legend at the bottom
    theme(legend.position = "bottom") +
    # Ensure that the legend is horizontal
    guides(color = guide_legend(nrow = 1, byrow = TRUE),
           shape = guide_legend(nrow = 1, byrow = TRUE))

  # d) To Print/Display the plot in RStudio
  print(combined_trend_plot)

  # e) Save the plot as an image file
  ggsave("combined_trend_plot_2023.png", plot = combined_trend_plot, width = 12, height = 6, dpi = 300)
}
# f) Generate plot
generate_combined_trend_plot(all_data)


```

a) **Filtering Data for 2023**: The data is filtered to only include records from the year 2023. This ensures that the trend analysis is specific to that year.

b) **Creating a Combined Trend Plot**:

- **geom_line**: This function is used to plot trend lines for each city's PM2.5 levels, averaging the data by month.
- **geom_point**: This adds individual data points to the plot, providing a granular view of the data alongside the trend lines.
- **scale_shape_manual**: Allows for custom shapes to be used for the points representing different cities, aiding visual distinction between them.

c) **Customizing the Plot**:

- **theme_minimal**: Applies a minimalistic theme to the plot for a clean data.
- **labs**: Adds labels and titles to the plot, such as the title of the plot and labels for the axes and legend.
- **theme and guides**: These functions adjust the positioning and layout of the legend, ensuring it is at the bottom and horizontal.

d) **Displaying and Saving the Plot**:

- The plot is printed out in the RStudio environment, allowing the user to visually inspect it.
- **ggsave**: This function saves the created plot as an image file on the user's computer, specifying dimensions and resolution.

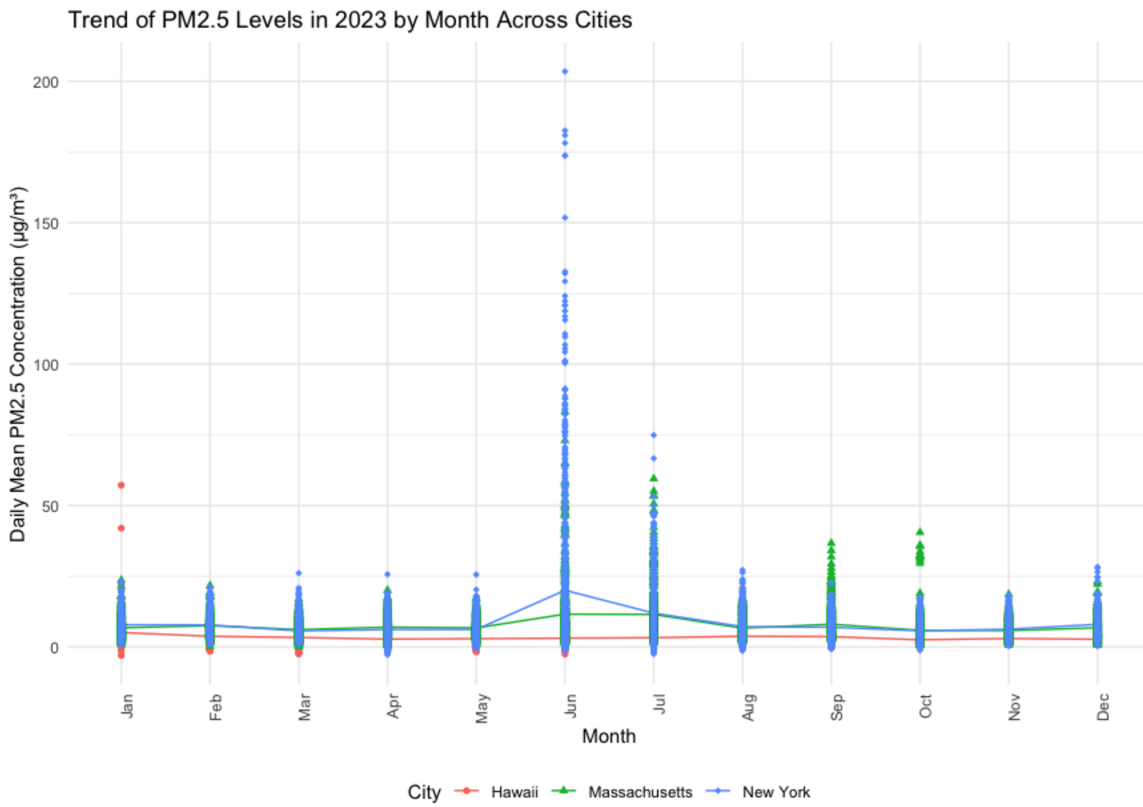
e) **Executing the Function**:

- Finally, the function **generate_combined_trend_plot** is called with the combined dataset **all_data**, which executes all the above steps to produce and save the trend plot.

For *additional information about the functions used in this script, please refer to the following hyperlinks:*

- geom_line
- geom_point
- scale_shape_manual
- theme_minimal
- labs
- theme
- guides
- ggsave

See below the plot generated by the script:



The plot displays the trend of PM2.5 levels across three cities—Hawaii, Massachusetts, and New York—over each month in the year 2023. The individual data points for each city are plotted monthly, with a line representing the mean trend of PM2.5 concentrations.

From the plot, you can observe that while there are fluctuations in PM2.5 levels throughout the year for each city, New York and Massachusetts exhibit spikes indicating higher pollution events, especially pronounced in New York. Hawaii's data points show less variation, suggesting more stable air quality. The trend lines help to smooth out this variability and provide a clearer picture of the overall trend across the year, which can be useful for identifying patterns or changes in air quality over time.

The final part of this guide underscores the value of this simulation in predicting air quality trends and its application in environmental data analysis. By following this guide, you'll not only perform the analysis but also grasp the reasoning behind each step.

End of Manual Thank you !!!!

References:

Packages links:

https://cran.r-project.org/web/packages/scales/index.html

https://cran.r-project.org/web/packages/ggplot2/index.html

https://cran.r-project.org/web/packages/dplyr/index.html

https://cran.r-project.org/web/packages/lubridate/index.html

https://cran.r-project.org/web/packages/readr/index.html

Functions used in the manual:

R library() function documentation

https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/function

https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/as.Date

https://lubridate.tidyverse.org/reference/year.html

https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/factor

https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/as.Date

https://lubridate.tidyverse.org/reference/year.html

https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/factor

https://r-graph-gallery.com/239-custom-layout-legend-ggplot2.html

https://ggplot2.tidyverse.org/

https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/ifelse

https://ggplot2.tidyverse.org/reference/ggsave.html

Data source:

https://www.epa.gov/outdoor-air-quality-data/download-daily-data

Air quality threshold:

https://www.c40knowledgehub.org/s/article/WHO-Air-Quality-Guidelines?language=en_US#:~:text=The%20current%20guidelines%20state%20that,3%20%2D%204%20days%20per%20year.