Rproject- Air Quality and Health Impact Analysis

ANUSHRI AHIR (22BDS0394)

2025-10-25

Air Quality and Health Impact Analysis — Lab Assessment Report

1. Dataset Description Source of Dataset: The dataset used in this project is sourced from the World Health Organization (WHO) Global Health Observatory under the Ambient Air Pollution (PM2.5) Dataset, 2022. It was downloaded as a CSV file and imported into R for analysis.

Description: The dataset contains information on fine particulate matter (PM2.5) concentrations across different countries, residence types (City, Rural, Urban, Total), and regions for multiple years. It is a reliable global dataset that helps study air quality and its potential health impact.

Structure:

Number of Records: 9,450 Number of Columns: 34

Main Attributes Used:

Region: Geographic grouping (e.g., Africa, Americas, Europe) Country: Country name Residence Type: Cities, Rural, Urban, Total Year (Period): Year of measurement FactValueNumeric (PM2.5): Annual mean concentration (µg/m³) Derived Fields: Health Impact Score, Health Category, Latitude, Longitude

Type of Dataset:

Format: CSV (Real-world dataset) Source: WHO Global Health Observatory Purpose: To analyze global air quality and assess its health impact through interactive visualizations.

2. Code Implementation

Below is an overview of the code implemented for the interactive visualizations. The full code was written and executed in R Markdown, using Plotly, Leaflet, and Rbokeh libraries.

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.3.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##
## 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
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.3
```

```
library(plotly)
```

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

```
##
## Attaching package: 'plotly'
```

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

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(leaflet)
library(RColorBrewer)
library(countrycode)
```

```
## Warning: package 'countrycode' was built under R version 4.3.3
```

```
library(rnaturalearth)
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.3.3
```

```
## Linking to GEOS 3.11.2, GDAL 3.8.2, PROJ 9.3.1; sf_use_s2() is TRUE
```

```r
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(DT)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)
```

```r
data_path <- "data (2).csv"
stopifnot(file.exists(data_path)) # stops if file not found
```

```r
getwd()
```

```
## [1] "C:/Users/LENOVO/Documents/SEM07/pds"
```

```r
#Load and quick inspect

raw <- read_csv(data_path, guess_max = 5000)
glimpse(raw)
```

```
## Rows: 9,450
## Columns: 34
## $ IndicatorCode            <chr> "SDGPM25", "SDGPM25", "SDGPM25", "SDGPM25",…
## $ Indicator                <chr> "Concentrations of fine particulate matter …
## $ ValueType                <chr> "text", "text", "text", "text", "text", "te…
## $ ParentLocationCode       <chr> "AFR", "AMR", "EUR", "AMR", "AMR", "EUR", "…
## $ ParentLocation           <chr> "Africa", "Americas", "Europe", "Americas",…
## $ `Location type`          <chr> "Country", "Country", "Country", "Country",…
## $ SpatialDimValueCode      <chr> "KEN", "TTO", "GBR", "GRD", "BRA", "DNK", "…
## $ Location                 <chr> "Kenya", "Trinidad and Tobago", "United Kin…
## $ `Period type`            <chr> "Year", "Year", "Year", "Year", "Year", "Ye…
## $ Period                   <dbl> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2…
## $ IsLatestYear             <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, T…
## $ `Dim1 type`              <chr> "Residence Area Type", "Residence Area Type…
## $ Dim1                     <chr> "Cities", "Rural", "Cities", "Total", "Town…
## $ Dim1ValueCode            <chr> "RESIDENCEAREATYPE_CITY", "RESIDENCEAREATYP…
## $ `Dim2 type`              <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ Dim2                     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ Dim2ValueCode            <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ `Dim3 type`              <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ Dim3                     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ Dim3ValueCode            <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ DataSourceDimValueCode   <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ DataSource               <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ FactValueNumericPrefix   <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ FactValueNumeric         <dbl> 10.01, 10.02, 10.06, 10.08, 10.09, 10.12, 1…
## $ FactValueUoM             <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ FactValueNumericLowPrefix <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ FactValueNumericLow      <dbl> 6.29, 7.44, 9.73, 7.07, 8.23, 9.37, 8.58, 9…
## $ FactValueNumericHighPrefix <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ FactValueNumericHigh     <dbl> 13.74, 12.55, 10.39, 13.20, 12.46, 10.97, 1…
## $ Value                    <chr> "10.01 [6.29-13.74]", "10.02 [7.44-12.55]",…
## $ FactValueTranslationID   <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ FactComments             <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
## $ Language                 <chr> "EN", "EN", "EN", "EN", "EN", "EN", "EN", "…
## $ DateModified             <dttm> 2022-08-11 18:30:00, 2022-08-11 18:30:00, …
```

```
DT::datatable(head(raw, 8), options = list(scrollX = TRUE))#show header sample
```

Show 10 entries Search:

	IndicatorCode	Indicator	ValueType	ParentLocationCode	ParentLocatior
1	SDGPM25	Concentrations of fine particulate matter (PM2.5)	text	AFR	Africa

2	SDGPM25	Concentrations of fine particulate matter (PM2.5)	text	AMR	Americas
3	SDGPM25	Concentrations of fine particulate matter (PM2.5)	text	EUR	Europe
4	SDGPM25	Concentrations of fine particulate matter (PM2.5)	text	AMR	Americas
5	SDGPM25	Concentrations of fine particulate matter (PM2.5)	text	AMR	Americas
6	SDGPM25	Concentrations of fine particulate matter (PM2.5)	text	EUR	Europe
7	SDGPM25	Concentrations of fine particulate matter (PM2.5)	text	EUR	Europe
8	SDGPM25	Concentrations of fine particulate matter (PM2.5)	text	EUR	Europe

Showing 1 to 8 of 8 entries

Previous 1 Next

```
#Canonicalize columns

df <- raw %>%
rename(
Country = Location,
Region = ParentLocation,
ResidenceType = Dim1,
Year = Period,
PM25 = FactValueNumeric
) %>%
select(Country, Region, ResidenceType, Year, PM25, everything())#keep only rows with PM2
5 numeric
df <- df %>% filter(!is.na(PM25))
#quick checks

cat("Rows after filtering PM2.5:", nrow(df), "\n")
```

```
## Rows after filtering PM2.5: 9450
```

```
summary(df$PM25)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4.59   11.92   19.57   23.54   30.98   97.49
```

```
DT::datatable(head(df %>% select(Country, Region, ResidenceType, Year, PM25), 20), optio
ns = list(scrollX = TRUE))
```

Show 10 **entries** **Search:**

	Country	Region	ResidenceT		
1	Kenya	Africa	Cities	2019	10.01
2	Trinidad and Tobago	Americas	Rural	2019	10.02
3	United Kingdom of Great Britain and Northern Ireland	Europe	Cities	2019	10.06
4	Grenada	Americas	Total	2019	10.08
5	Brazil	Americas	Towns	2019	10.09
6	Denmark	Europe	Urban	2019	10.12
7	Russian Federation	Europe	Cities	2019	10.19
8	Spain	Europe	Cities	2019	10.19
9	Grenada	Americas	Towns	2019	10.22

10	Grenada	Americas	Urban	2019	10.22

Showing 1 to 10 of 20 entries Previous 1 2 Next

```
#Derive HealthImpactScore and HealthCategory

df <- df %>%
mutate(
# numeric Year

Year = as.integer(Year),
HealthImpactScore = case_when(
PM25 <= 5 ~ 1,
PM25 <= 15 ~ 2,
PM25 <= 25 ~ 3,
PM25 <= 35 ~ 4,
TRUE ~ 5
),
HealthCategory = case_when(
HealthImpactScore == 1 ~ "Low",
HealthImpactScore == 2 ~ "Moderate",
HealthImpactScore == 3 ~ "Sensitive groups",
HealthImpactScore == 4 ~ "Unhealthy",
HealthImpactScore == 5 ~ "Hazardous"
)
)
table(df$HealthCategory)
```

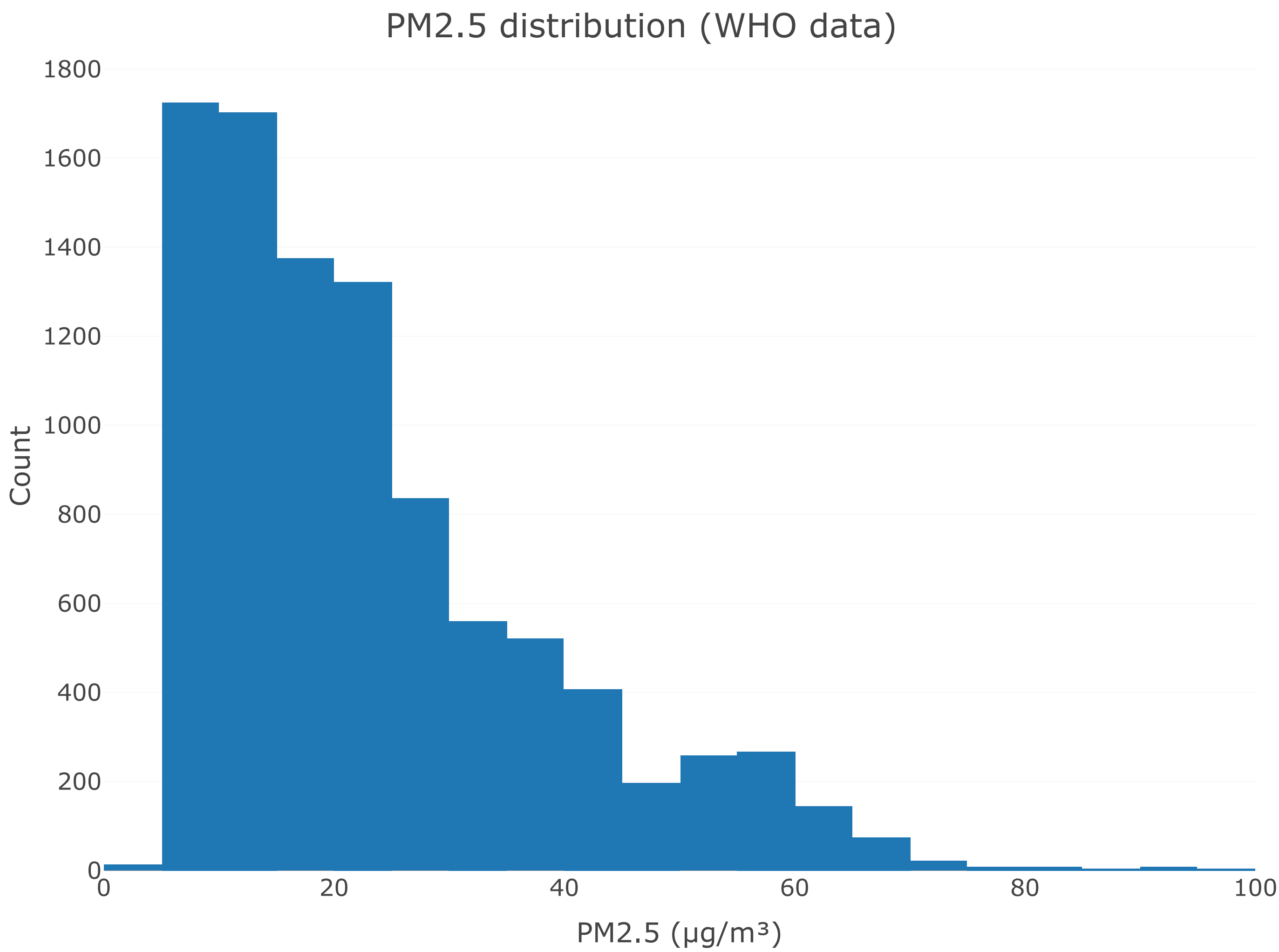
```
##
##        Hazardous              Low         Moderate Sensitive groups
##             1911               15             3429             2697
##        Unhealthy
##             1398
```

```
# Install required packages if missing

if (!require("rnaturalearth")) install.packages("rnaturalearth")
if (!require("rnaturalearthdata")) install.packages("rnaturalearthdata")
if (!require("sf")) install.packages("sf")
if (!require("countrycode")) install.packages("countrycode")

library(rnaturalearth)
library(rnaturalearthdata)
library(sf)
library(countrycode)
library(dplyr)

# Add country ISO3 codes
df <- df %>%
  mutate(iso3 = countrycode(Country, "country.name", "iso3c"))

# Load world map safely
world <- rnaturalearth::ne_countries(scale = "medium", returnclass = "sf")

# Compute centroids for each country polygon
centroids <- st_centroid(world) %>%
  cbind(st_coordinates(.)) %>%
  select(iso_a3, X, Y) %>%
  rename(iso3 = iso_a3, long = X, lat = Y)

# Merge with main dataframe
df <- df %>%
  left_join(centroids, by = "iso3")

# Handle missing coordinates with random jitter (for countries not found)
missing_geo <- which(is.na(df$lat) | is.na(df$long))
if (length(missing_geo) > 0) {
  set.seed(42)
  df$lat[missing_geo] <- runif(length(missing_geo), -10, 50)
  df$long[missing_geo] <- runif(length(missing_geo), -80, 100)
}

# Check if coordinates now exist
head(df %>% select(Country, lat, long))
```

```
## # A tibble: 6 × 3
##   Country                                                lat   long
##   <chr>                                                <dbl>  <dbl>
## 1 Kenya                                                0.599  37.8
## 2 Trinidad and Tobago                                 10.5   -61.3
## 3 United Kingdom of Great Britain and Northern Ireland  54.0    -2.76
## 4 Grenada                                             12.1   -61.7
## 5 Brazil                                             -10.6   -53.2
## 6 Denmark                                             56.0    10.0
```

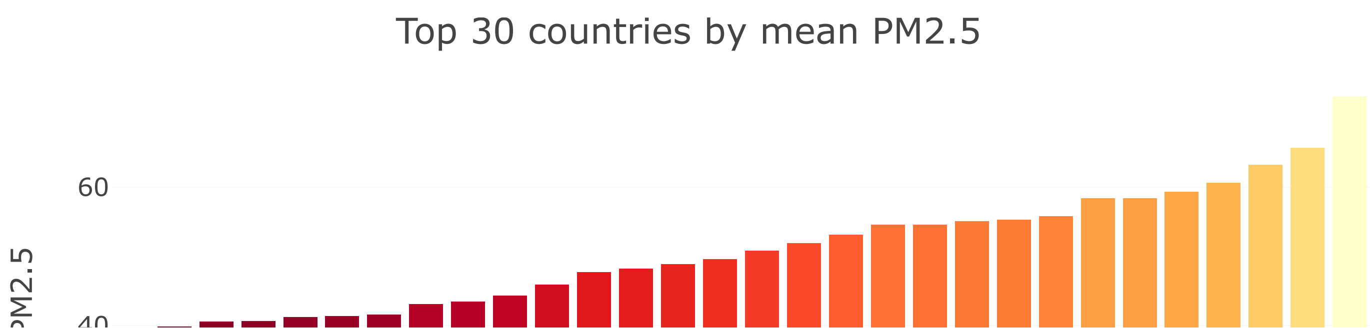
Visualizations:

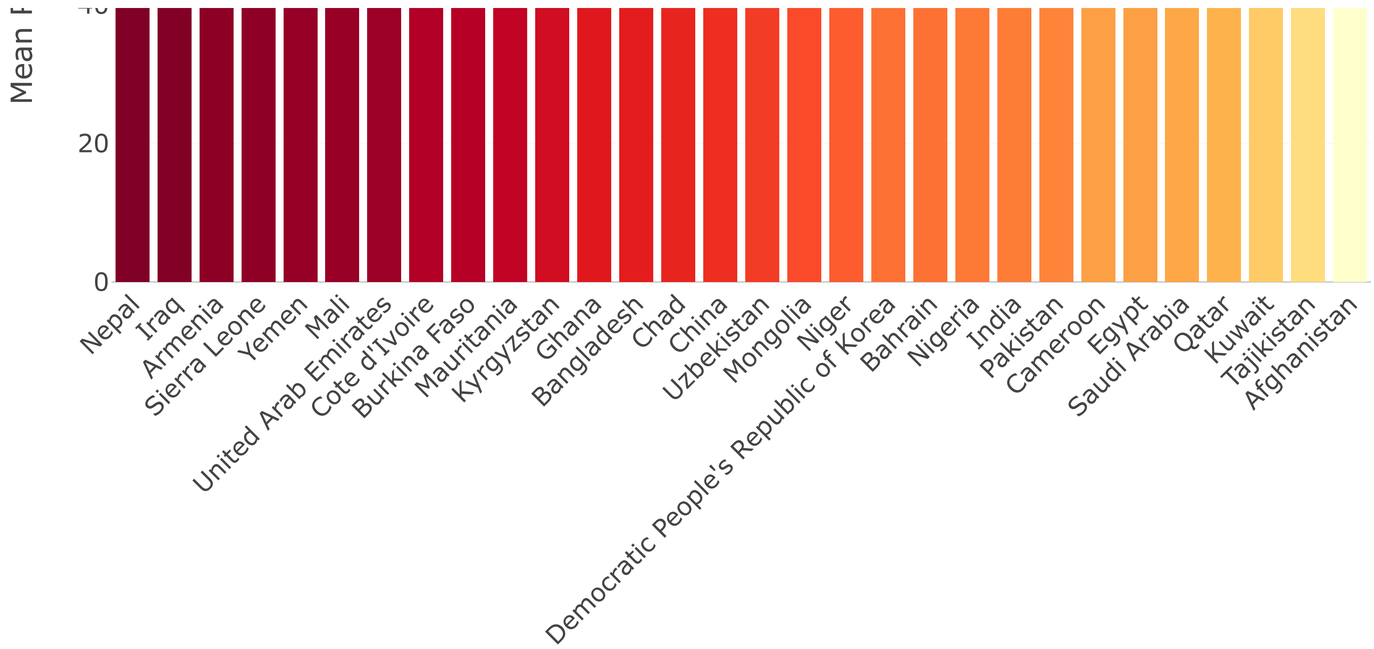
```
#1.Histogram – PM2.5 distribution

plot_ly(df, x=~PM25, type='histogram', nbinsx = 40) %>%
layout(title = "PM2.5 distribution (WHO data)", xaxis=list(title="PM2.5 (µg/m³)"), yaxis
=list(title="Count"))
```



Insight: Shows how global PM2.5 values are distributed. Most countries fall under low exposure, with a long left tail representing low-pollution zones.

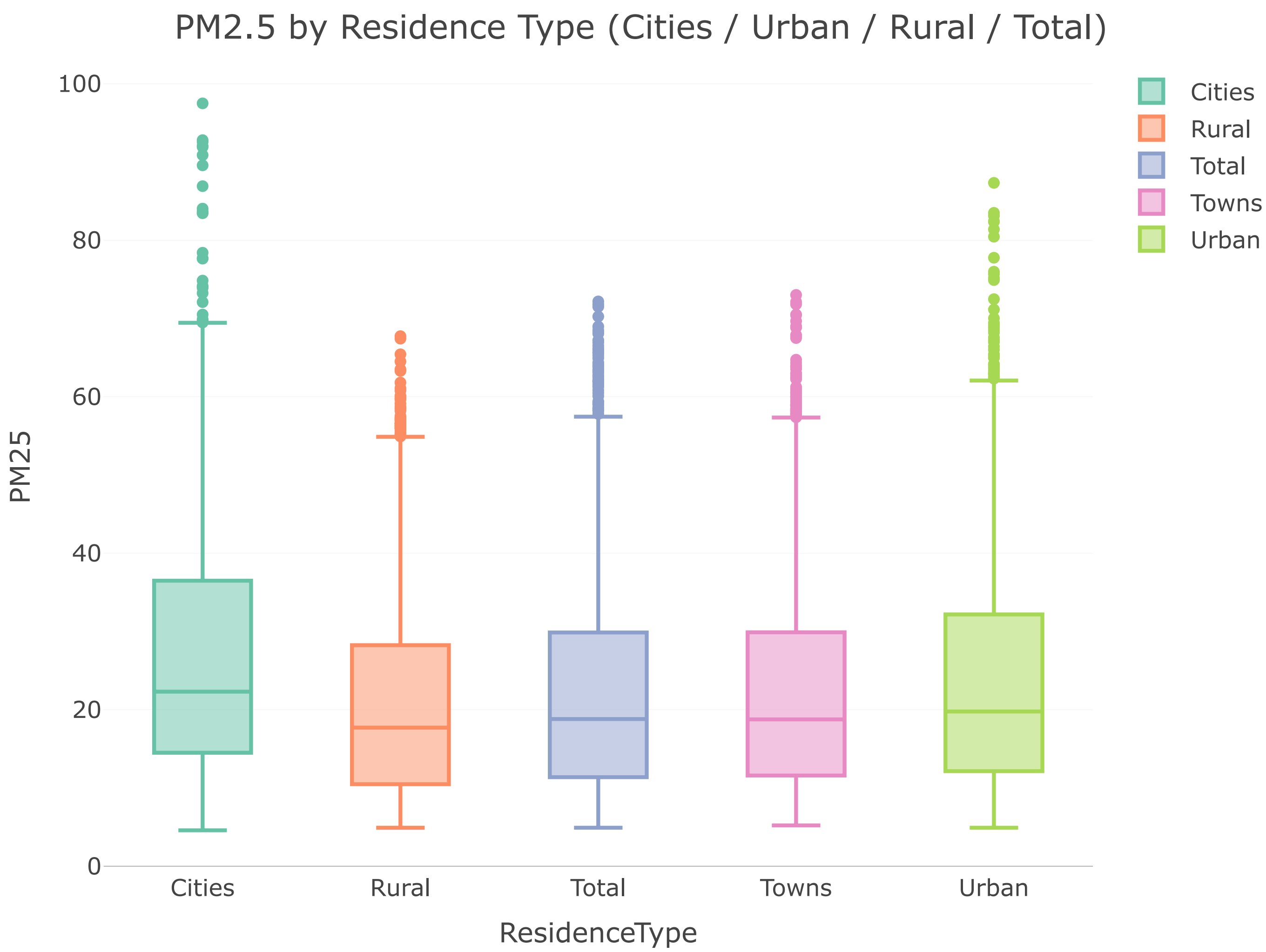
```
#2.Top 30 countries by mean PM2.5 (bar chart)

top_countries <- df %>% group_by(Country) %>% summarise(mean_PM25 = mean(PM25, na.rm=TRU
E), n = n()) %>% arrange(desc(mean_PM25)) %>% slice(1:30)
plot_ly(top_countries, x=~reorder(Country, mean_PM25), y=~mean_PM25, type='bar', marker=
list(color=~mean_PM25, colorscale='YlOrRd')) %>%
layout(title="Top 30 countries by mean PM2.5", xaxis = list(title="", tickangle=-45), ya
xis=list(title="Mean PM2.5"))
```





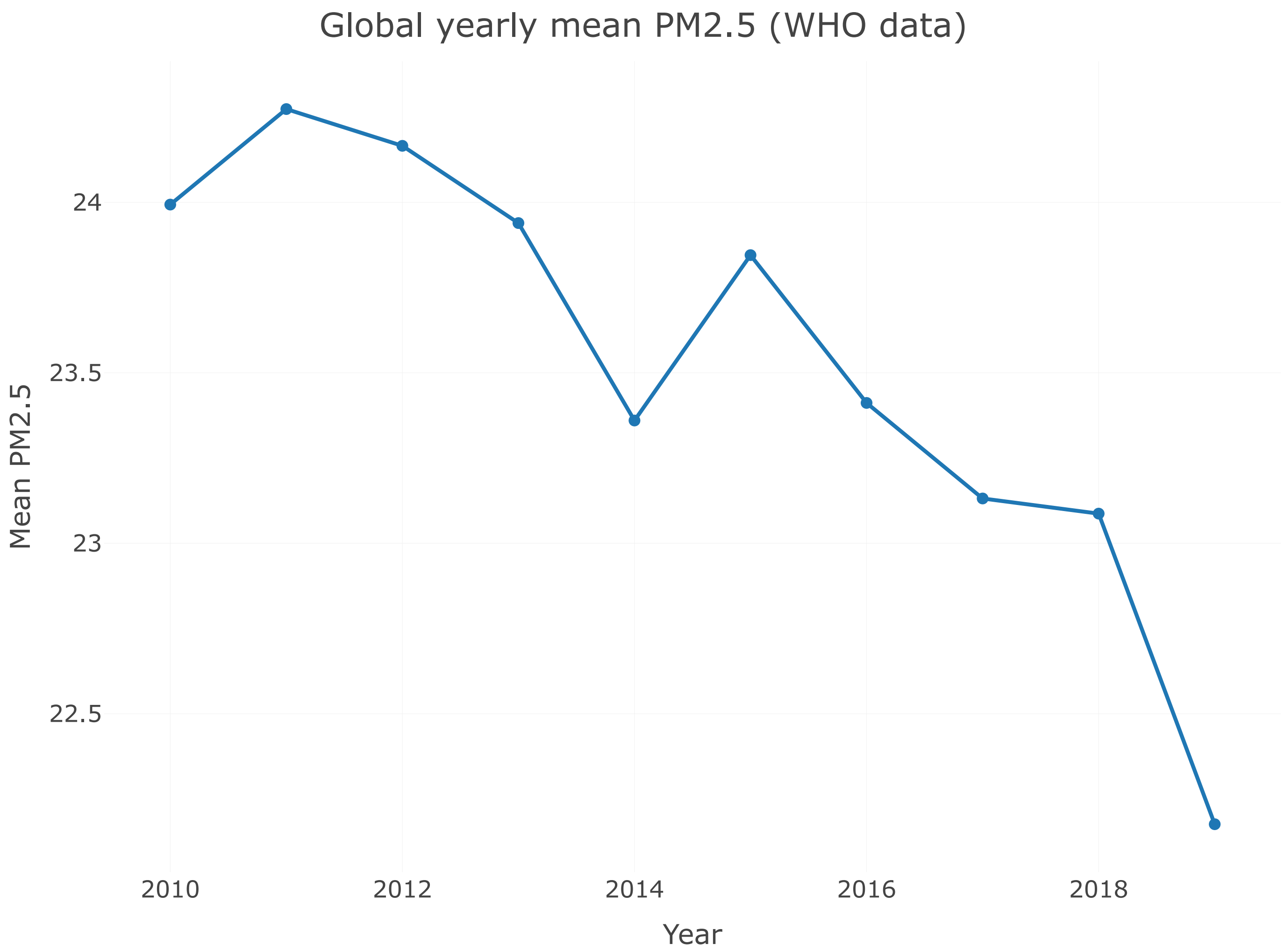
Insight: Highlights the most polluted countries globally, using YlOrRd color scale for intuitive severity display.

```
#3.PM2.5 by Residence Type (boxplot)

plot_ly(df, x=~ResidenceType, y=~PM25, type='box', color=~ResidenceType) %>%
layout(title="PM2.5 by Residence Type (Cities / Urban / Rural / Total)")
```



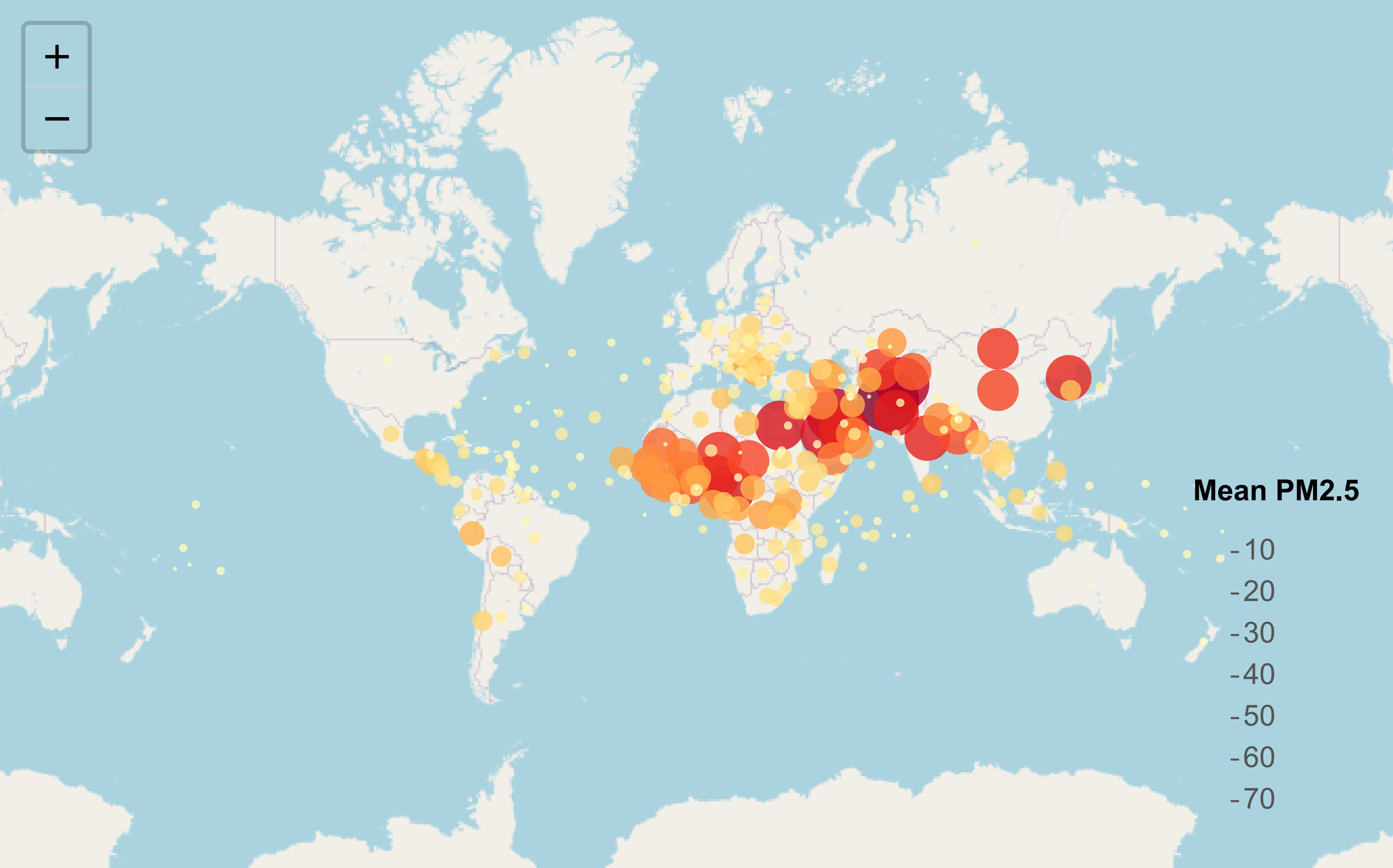
Insight: City and urban areas show higher median pollution levels.

```
#4.PM2.5 time trend (Yearly mean) – if multiple years present

if(sum(!is.na(df$Year))>0){
yearly <- df %>% group_by(Year) %>% summarise(mean_PM25 = mean(PM25, na.rm=TRUE))
plot_ly(yearly, x=~Year, y=~mean_PM25, type='scatter', mode='lines+markers') %>%
layout(title="Global yearly mean PM2.5 (WHO data)", xaxis=list(title="Year"), yaxis=list
(title="Mean PM2.5"))
} else { cat("Year information not available for trend plot.") }
```



Insight: Displays global PM2.5 trends over time. Used to assess if pollution levels have decreased or stabilized in recent years.

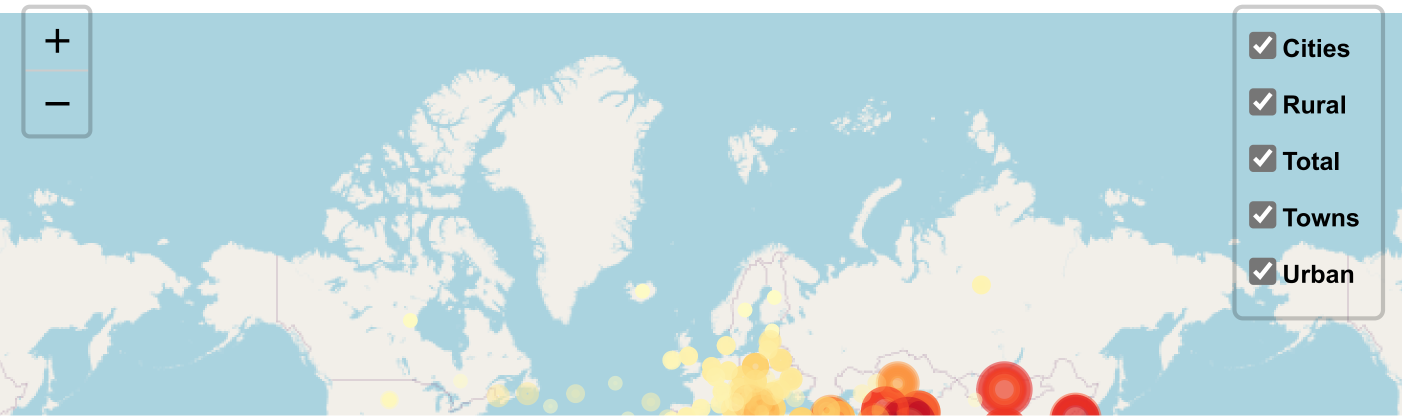
```
#5.Map – mean PM2.5 by country (Leaflet)

map_data <- df %>% group_by(Country, lat, long) %>% summarise(mean_PM25 = mean(PM25, na.
rm=TRUE), n = n()) %>% arrange(desc(mean_PM25))
pal <- colorNumeric("YlOrRd", domain = map_data$mean_PM25)
leaflet(map_data) %>% addTiles() %>%
addCircleMarkers(~long, ~lat, radius = ~pmin(15, mean_PM25/5), color = ~pal(mean_PM25),
stroke = FALSE, fillOpacity = 0.8,
popup = ~paste0("", Country, "Mean PM2.5: ", round(mean_PM25,1))) %>%
addLegend("bottomright", pal = pal, values = ~mean_PM25, title = "Mean PM2.5")
```

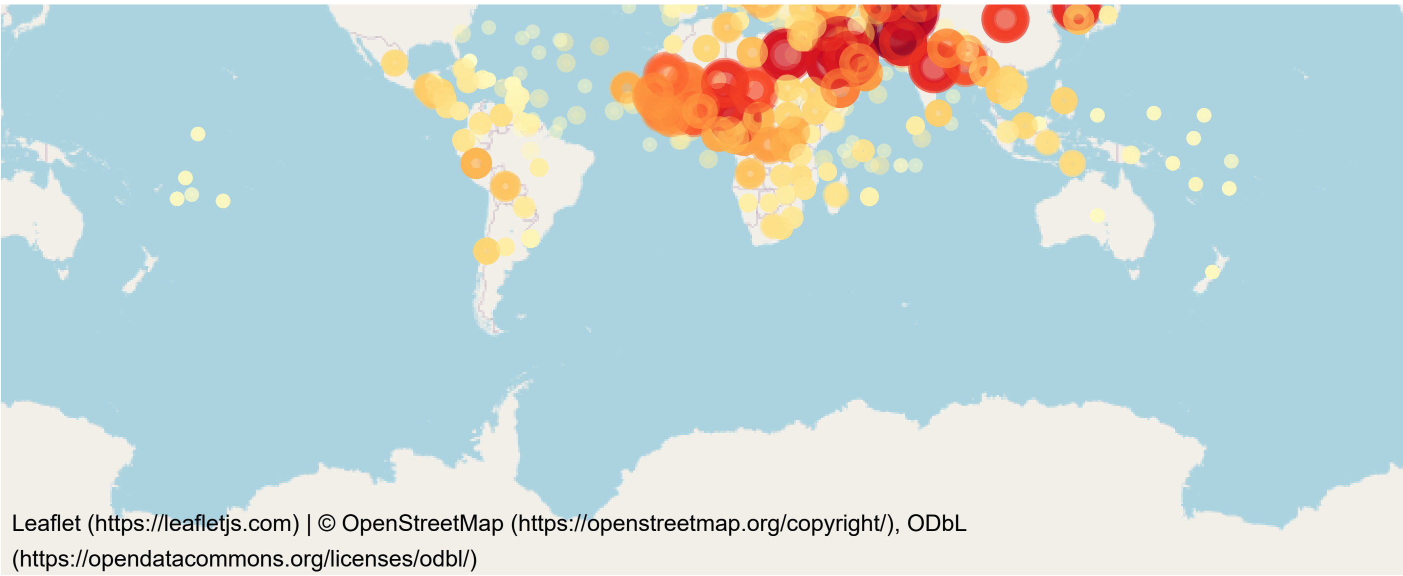


Leaflet (https://leafletjs.com) | © OpenStreetMap (https://openstreetmap.org/copyright/), ODbL (https://opendatacommons.org/licenses/odbl/)

Insight: Provides a spatial overview of pollution hotspots; larger red markers indicate higher PM2.5.

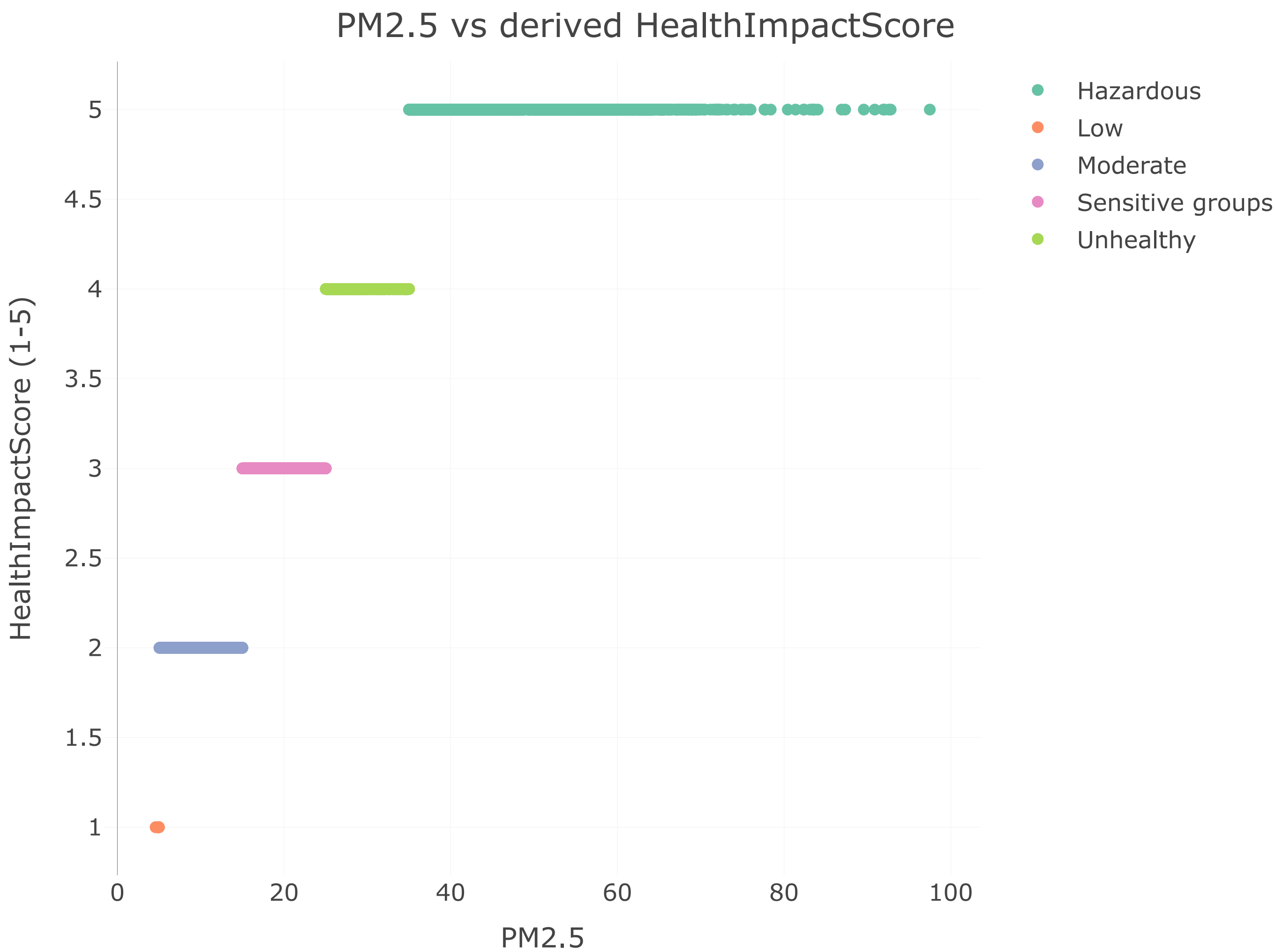
```
#6.Leaflet multi-layer view (ResidenceType filter via groups)

leaflet() %>% addTiles() %>%
{ m <- .
for(rt in unique(na.omit(df$ResidenceType))){
sub <- df %>% filter(ResidenceType == rt) %>% group_by(Country, lat, long) %>% summarise
(mean_PM25 = mean(PM25, na.rm=TRUE))
m <- m %>% addCircleMarkers(data=sub, lng=~long, lat=~lat, radius=~pmin(12, mean_PM25/
6), color=~colorNumeric("YlOrRd", sub$mean_PM25)(sub$mean_PM25), group = rt,
popup = ~paste0("", Country, "Residence: ", rt, "Mean PM2.5: ", round(mean_PM25,1)))
}
m %>% addLayersControl(overlayGroups = unique(na.omit(df$ResidenceType)), options = laye
rsControlOptions(collapsed=FALSE))
}
```





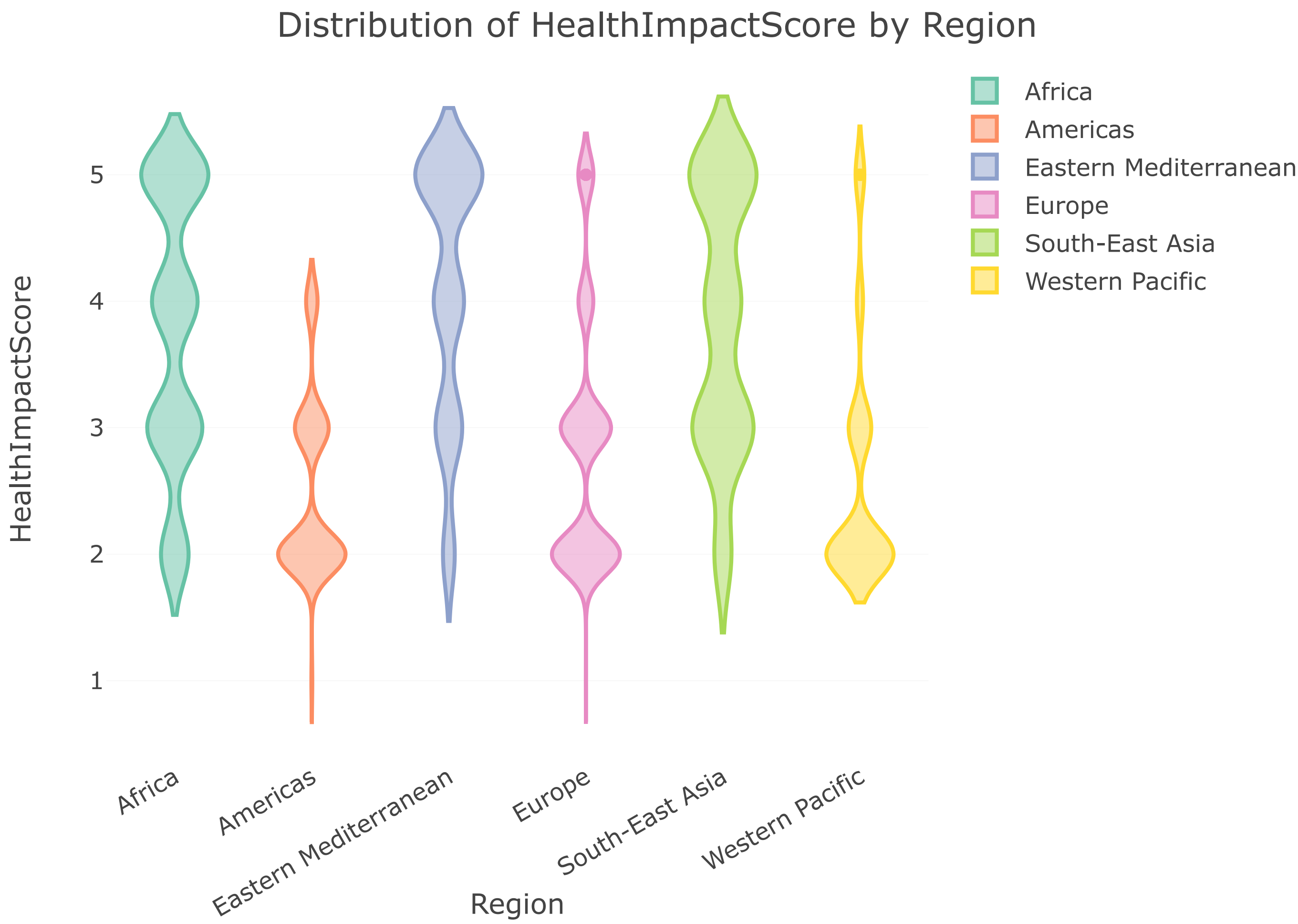
Insight: Enables filtering of pollution levels by residence category — interactive comparison between population zones.

```
#7.PM2.5 vs HealthImpactScore (scatter plot)

plot_ly(df, x=~PM25, y=~HealthImpactScore, color=~HealthCategory, text=~paste("Countr
y:", Country, "Res:", ResidenceType), type='scatter', mode='markers') %>%
layout(title="PM2.5 vs derived HealthImpactScore", xaxis=list(title="PM2.5"), yaxis=list
(title="HealthImpactScore (1-5)"))
```



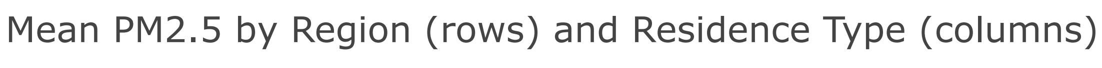
Insight: Clear positive correlation between PM2.5 and health risk.

```
#8.Violin / distribution of HealthImpactScore by Region

plot_ly(df, x=~Region, y=~HealthImpactScore, type='violin', color=~Region) %>%
layout(title="Distribution of HealthImpactScore by Region", xaxis=list(tickangle=-30))
```



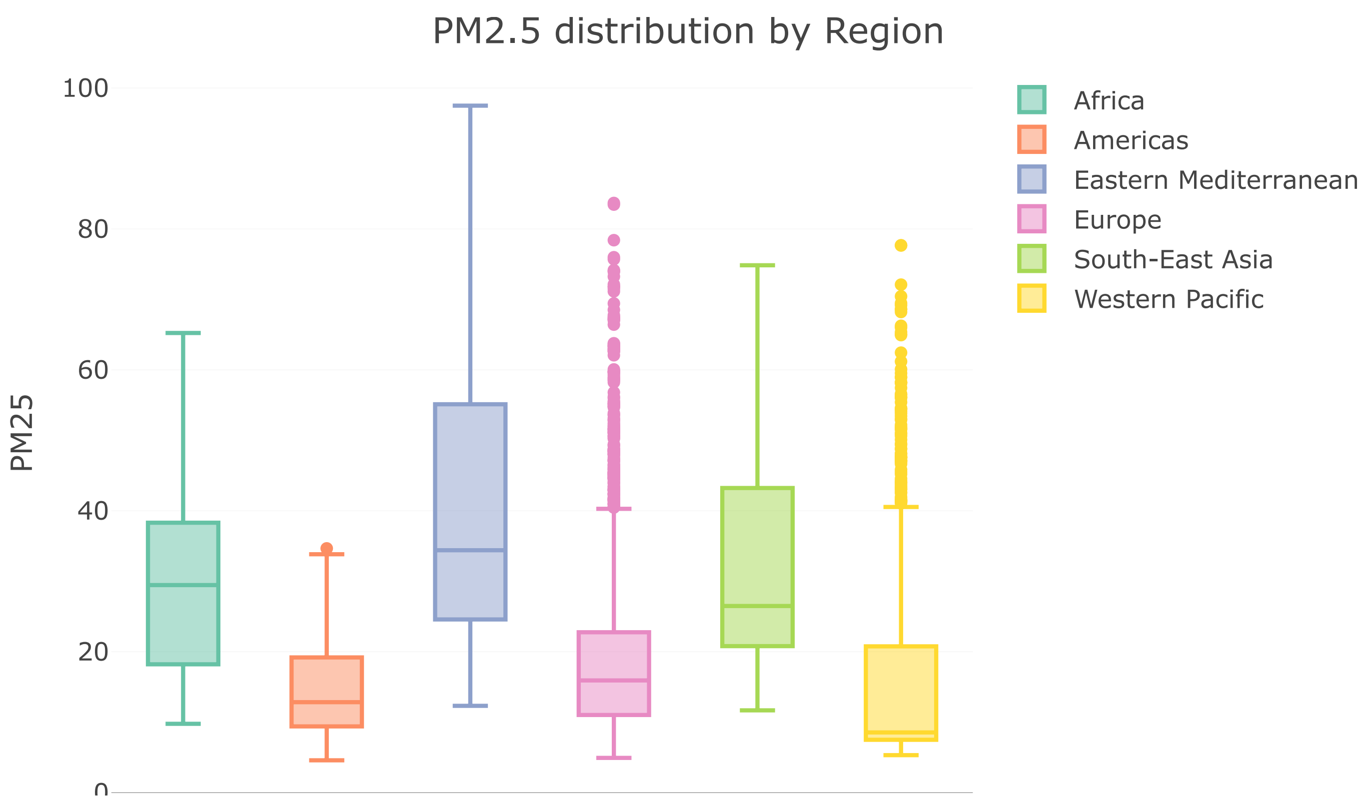
Insight: Demonstrates variability in health impact distribution across regions. Regions like Africa, south-east Asia show broader, higher-risk distributions.

```
#9.Heatmap: Region × ResidenceType mean PM2.5

heat_df <- df %>% group_by(Region, ResidenceType) %>% summarise(mean_PM25 = mean(PM25, n
a.rm=TRUE)) %>% ungroup()

#pivot to matrix for plotly heatmap

heat_wide <- heat_df %>% pivot_wider(names_from = ResidenceType, values_from = mean_PM2
5, values_fill = 0)
zmat <- as.matrix(heat_wide %>% select(-Region))
plot_ly(x = colnames(zmat), y = heat_wide$Region, z = zmat, type = "heatmap", colorscale
= "Viridis") %>%
layout(title="Mean PM2.5 by Region (rows) and Residence Type (columns)", xaxis=list(titl
e="Residence Type"), yaxis=list(title="Region"))
```



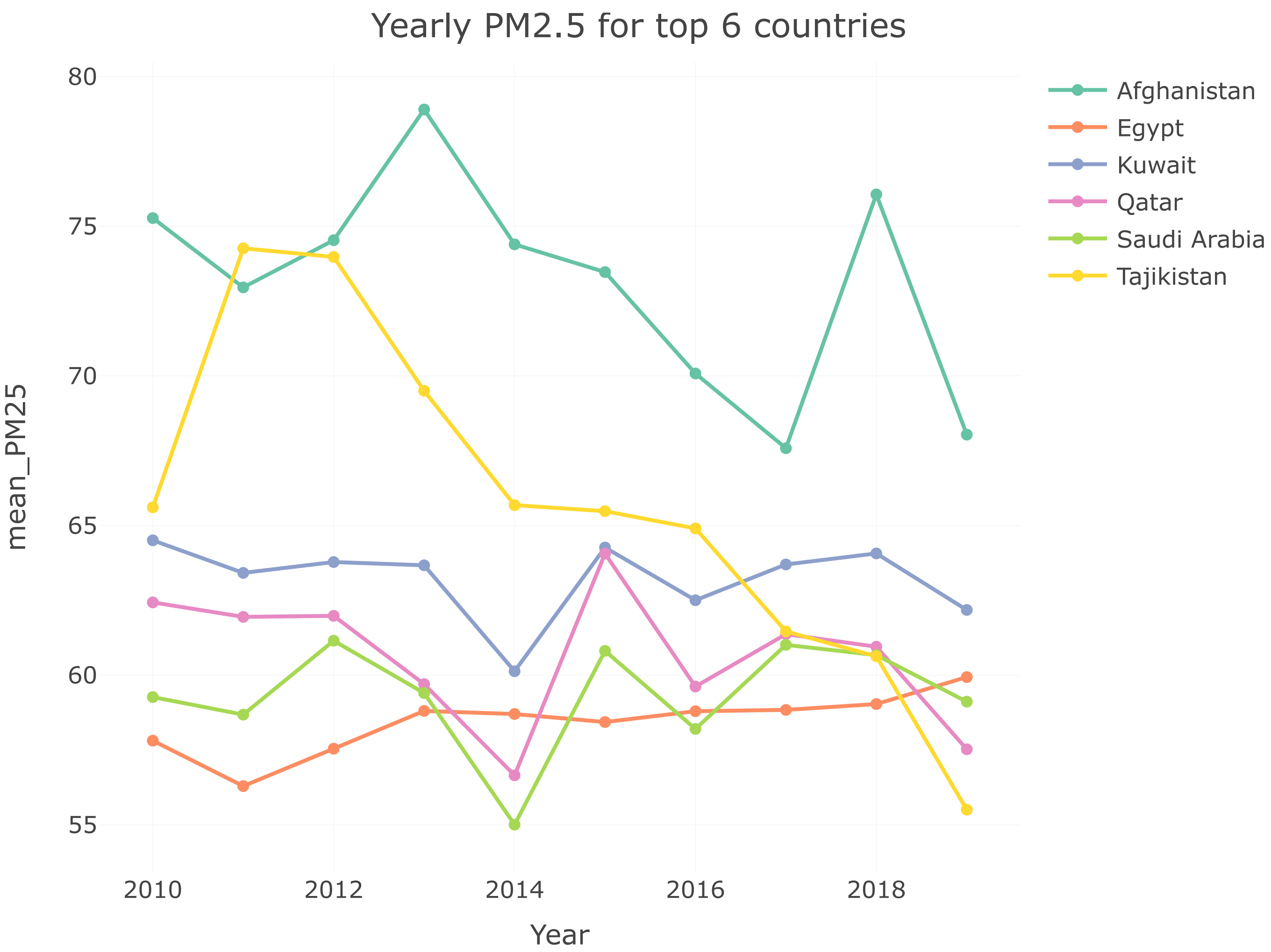
Insight: Reveals that cities and urban regions within certain continents contribute the most to pollution levels.

```
#10.Boxplot: PM2.5 by Region

plot_ly(df, x=~Region, y=~PM25, type='box', color=~Region) %>%
layout(title="PM2.5 distribution by Region", xaxis=list(tickangle=-30))
```



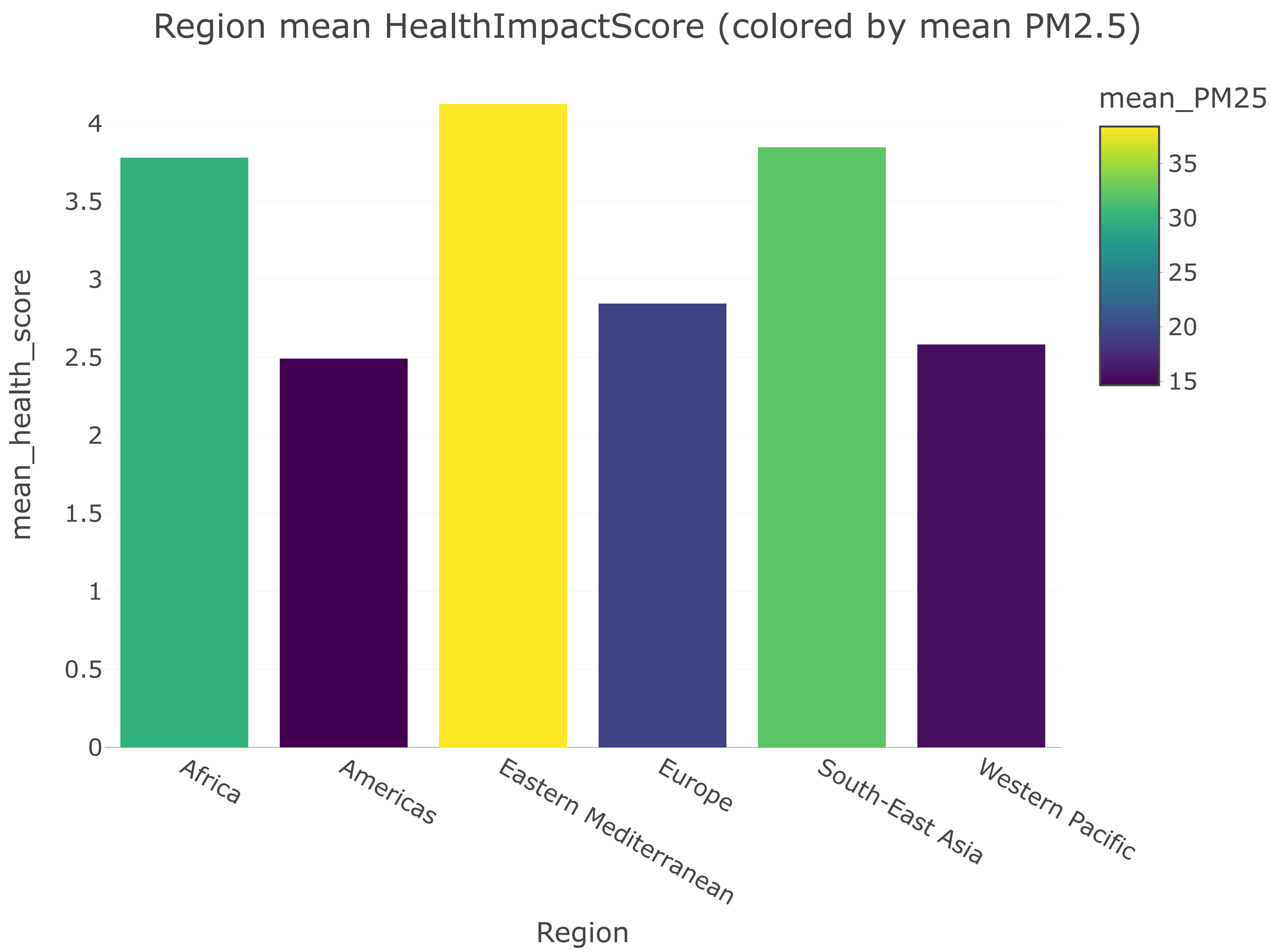
Insight: Confirms inter-regional disparity — Eastern Mediterranean and South-East Asia dominate the upper quartiles.

```
#11.Small multiples: 6-country yearly trends (if Year exists)

top6 <- top_countries %>% slice(1:6) %>% pull(Country)
cmp <- df %>% filter(Country %in% top6) %>% group_by(Country, Year) %>% summarise(mean_P
M25 = mean(PM25, na.rm=TRUE))
plot_ly(cmp, x=~Year, y=~mean_PM25, color=~Country, split=~Country, type='scatter', mode
='lines+markers') %>%
layout(title='Yearly PM2.5 for top 6 countries')
```



Insight: Shows if heavily polluted countries are improving or worsening over time.

```r
#12.Correlation-like summary: parameter influence (here PM2.5 only)

region_rank <- df %>% group_by(Region) %>% summarise(mean_health_score = mean(HealthImpa
ctScore, na.rm=TRUE), mean_PM25 = mean(PM25, na.rm=TRUE)) %>% arrange(desc(mean_health_s
core))
plot_ly(region_rank, x=~Region, y=~mean_health_score, type='bar', color=~mean_PM25, colo
rscale='YlOrRd') %>%
layout(title="Region mean HealthImpactScore (colored by mean PM2.5)")
```




```r
region_rank
```

```
## # A tibble: 6 × 3
##   Region                mean_health_score mean_PM25
##   <chr>                             <dbl>     <dbl>
## 1 Eastern Mediterranean              4.13      38.4
## 2 South-East Asia                    3.85      32.1
## 3 Africa                             3.78      29.8
## 4 Europe                             2.84      19.2
## 5 Western Pacific                    2.58      15.5
## 6 Americas                           2.49      14.6
```

Insight: Regions with high PM2.5 also show high average health impact scores — confirming PM2.5 as the dominant influencing parameter.

```
#13.Interactive data table for queries

DT::datatable(df %>% select(Country, Region, ResidenceType, Year, PM25, HealthCategory,
HealthImpactScore) %>% arrange(desc(PM25)),
options = list(pageLength = 10, scrollX = TRUE))
```

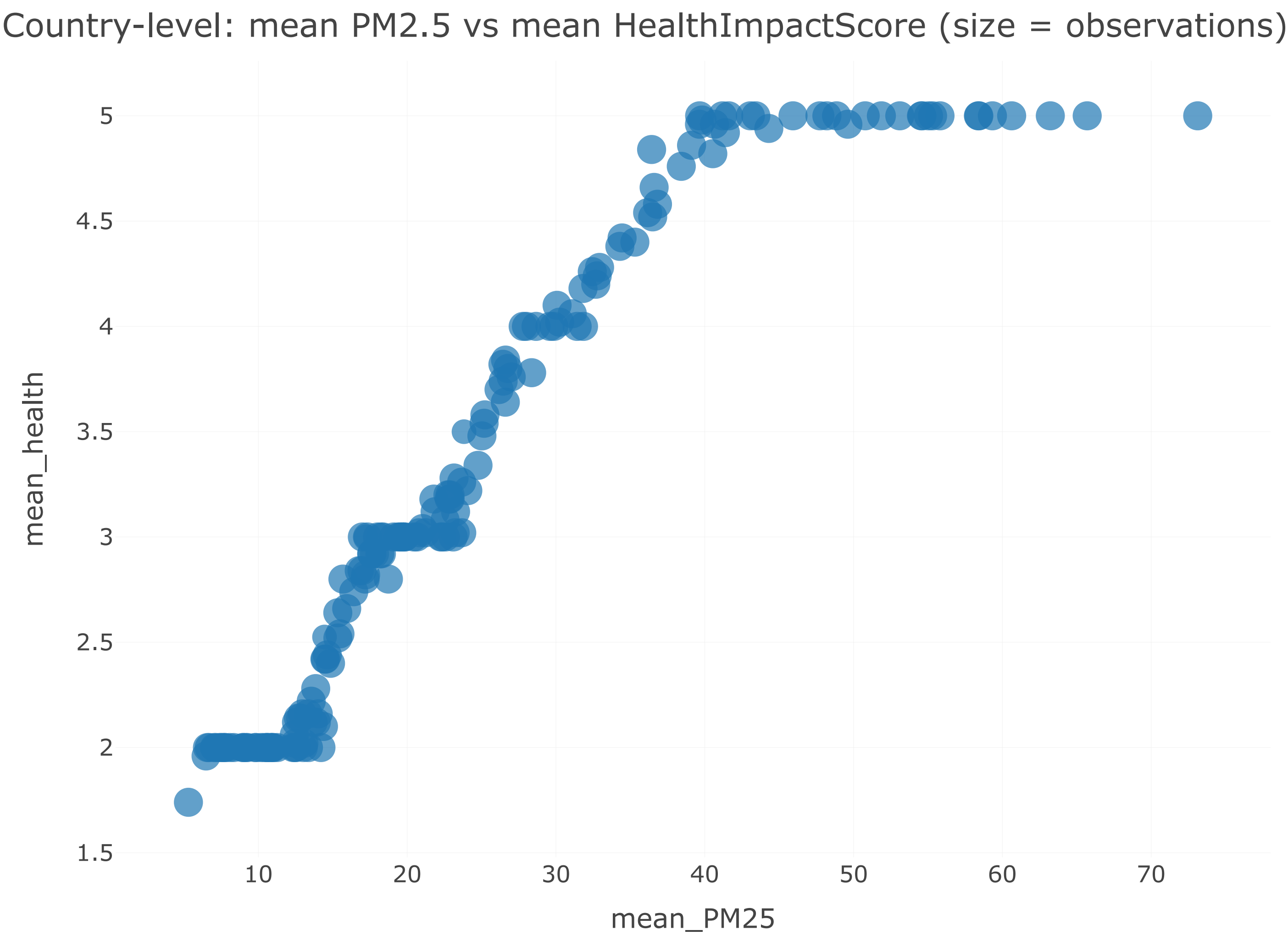
Show 10 **entries** **Search:**

	Country	Region	ResidenceType	Year	PM25	HealthC
1	Afghanistan	Eastern Mediterranean	Cities	2013	97.49	Hazardous
2	Afghanistan	Eastern Mediterranean	Cities	2010	92.79	Hazardous
3	Afghanistan	Eastern Mediterranean	Cities	2018	92.57	Hazardous
4	Afghanistan	Eastern Mediterranean	Cities	2012	92.04	Hazardous
5	Afghanistan	Eastern Mediterranean	Cities	2014	91.92	Hazardous
6	Afghanistan	Eastern Mediterranean	Cities	2015	90.88	Hazardous
7	Afghanistan	Eastern Mediterranean	Cities	2011	89.57	Hazardous
8	Afghanistan	Eastern Mediterranean	Urban	2013	87.33	Hazardous
9	Afghanistan	Eastern Mediterranean	Cities	2016	86.91	Hazardous
10	Afghanistan	Eastern Mediterranean	Cities	2019	84.04	Hazardous

Showing 1 to 10 of 9,450 entries Previous 1 2 3 4 5 … 945 Next

Insight: Allows users to query countries, filter regions, and directly view pollution-health relationships interactively.

```
#14.Bubble chart: HealthImpactScore vs mean_PM25 by country (size = count)

bubble <- df %>% group_by(Country) %>% summarise(mean_PM25 = mean(PM25, na.rm=TRUE), mea
n_health = mean(HealthImpactScore, na.rm=TRUE), n=n())
plot_ly(bubble, x=~mean_PM25, y=~mean_health, size=~n, text=~Country, type='scatter', mo
de='markers', marker=list(sizemode='area')) %>%
layout(title="Country-level: mean PM2.5 vs mean HealthImpactScore (size = observation
s)")
```



Insight: Visualizes both PM2.5 intensity and data frequency. Larger bubbles indicate more complete data; color and position reflect health severity

```
#Save cleaned dataset

write_csv(df %>%
            select(Country, Region, ResidenceType, Year, PM25, HealthCategory, HealthImp
actScore, lat, long),
          "who_pm25_cleaned.csv")
cat("Saved cleaned dataset to who_pm25_cleaned.csv\n")
```

```
## Saved cleaned dataset to who_pm25_cleaned.csv
```

```
# Read the CSV back into R
who_pm25_cleaned <- read_csv("who_pm25_cleaned.csv")
head(who_pm25_cleaned, 10)  # shows first 10 rows
```

```
## # A tibble: 10 × 9
##    Country     Region ResidenceType  Year  PM25 HealthCategory HealthImpactScore
##    <chr>       <chr>  <chr>         <dbl> <dbl> <chr>                      <dbl>
##  1 Kenya       Africa Cities         2019  10.0 Moderate                       2
##  2 Trinidad a… Ameri… Rural          2019  10.0 Moderate                       2
##  3 United Kin… Europe Cities         2019  10.1 Moderate                       2
##  4 Grenada     Ameri… Total          2019  10.1 Moderate                       2
##  5 Brazil      Ameri… Towns          2019  10.1 Moderate                       2
##  6 Denmark     Europe Urban          2019  10.1 Moderate                       2
##  7 Russian Fe… Europe Cities         2019  10.2 Moderate                       2
##  8 Spain       Europe Cities         2019  10.2 Moderate                       2
##  9 Grenada     Ameri… Towns          2019  10.2 Moderate                       2
## 10 Grenada     Ameri… Urban          2019  10.2 Moderate                       2
## # ℹ 2 more variables: lat <dbl>, long <dbl>
```

Click here to download the cleaned CSV (who_pm25_cleaned.csv)

4. Discussion of Results

Key Observations PM2.5 concentration is highest in urban and city zones. Africa, Eastern Mediterranean and South-East Asia display the most alarming PM2.5 averages. HealthImpactScore is directly correlated with PM2.5, making it the most critical influencing parameter. Interactive maps and plots revealed nuanced differences that static charts could not — especially the spatial relationships between air quality and health risk.

Parameter Influence PM2.5 concentration was the most influential factor determining health outcomes. The “Regional Correlation Summary” and “PM2.5 vs HealthImpactScore” plots both confirmed a near-linear rise in risk with increasing pollution levels.

Overall Insight: Interactive visualization revealed deeper patterns than static charts could — connecting pollution exposure, regional disparities, and health risk in one visual framework.

5. References

World Health Organization. (2022). Ambient Air Pollution Database (PM2.5). Retrieved from https://www.who.int/data/gho/data/themes/air-pollution (https://www.who.int/data/gho/data/themes/air-pollution) Plotly Technologies Inc. (2024). Plotly R Open Source Graphing Library. Retrieved from https://plotly.com/r/ (https://plotly.com/r/) Leaflet for R. (2024). Interactive Maps with R. RStudio. Retrieved from https://rstudio.github.io/leaflet/ (https://rstudio.github.io/leaflet/) Bokeh Developers. (2023). Rbokeh: An R Interface to Bokeh Visualization Library. Retrieved from https://hafen.github.io/rbokeh/ (https://hafen.github.io/rbokeh/) R Core Team. (2023). R: A Language and Environment for Statistical Computing. Vienna: R Foundation for Statistical Computing.